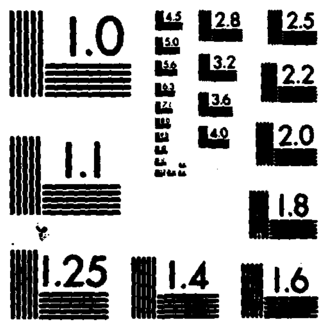
AD-A091 684 ARMY MISSILE COMMAND REDSTONE ARSENAL AL TECHNOLOGY--ETC F/G 5/1
A METHODOLOGY FOR AGGREGATION OF MULTIPLE CRITERIA RANK-ORDERED--ETC(U)
MAY 80 E G DOBBINS
UNCLASSIFIED DRSMI/RN-80-3-MR SBIE-AD-E950 034 NL

1 3



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL III ②

AD-E950 034

AD A091884

MANAGEMENT REPORT RN-80-3

A METHODOLOGY FOR AGGREGATION OF MULTIPLE CRITERIA RANK-ORDERED PRIORITIES

Edward B. Dobbins, Jr.
Technology Integration Office
US Army Missile Laboratory

May 1980

U.S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama 35809

DDC FILE COPY

Approved for public release; distribution unlimited

DTIC
SELECTED
NOV 24 1980
S D
B

80 11 24 008

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
RN-80-3

2. GOVT ACCESSION NO.
AD-A091 684

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
A Methodology for Aggregation of Multiple Criteria Rank-Ordered Priorities

5. TYPE OF REPORT & PERIOD COVERED
Management Report

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Edward B. Dobbins, Jr.

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Commander
US Army Missile Command
ATTN: DRSMI-RN
Redstone Arsenal, Alabama 35898

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Commander
US Army Missile Command
ATTN: DRSMI-RPT
Redstone Arsenal, Alabama 35898

12. REPORT DATE
May 1980

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Aggregation
Rank Orders
Research and Development
Prioritization
Fuzzy Set Rank Orders

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report is concerned with the aggregation of multiple rank-ordered research and development (R&D) projects or product requirements. The resultant prioritized list serves as a resource allocation basis for the R&D projects. The rank orders are ordinal and without feedback or strategy. The extensive literature of the social choice field is structured and analyzed for this research. Building upon that foundation, this research classified the known majority-rule aggregation methods, evaluated them, and developed a model and computer code

(continued)

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. using the better methods to compute the aggregation of up to 100 rank-ordered lists of up to 100 alternatives. Decision-maker and alternative weighting may be applied by any of eight methods. Judge self-evaluation weighting is also available. Final rank-ordered priority lists of R&D projects are computed by the Borda, Adjusted Borda, and Shannon Majority-Rule methods. The preferred Shannon method handles complete, partial, and weighted rank orders without transitivity. Fuzzy set rank orders are compared. Kendall's coefficients of consistency and concordance evaluate the rank orders. The model is validated by comparative computation of many rank-ordered aggregation examples from the literature.

ACKNOWLEDGMENTS

The author wishes to express his profound gratitude to his Ph.D. advisor, Dr. Robert E. Shannon, whose encouragement, helpful criticism, guidance, and assistance during the course of the research made this report possible. The author is also indebted to the other members of the committee, Dr. Robert A. Brown, Dr. Richard Wyskida, Dr. Norman J. Bucher, and Dr. Deva C. Doss for their advice and support in completion of this report.

Various US Army Missile Command personnel provided advice toward the research of this report. Special thanks are given to Mrs. Joyce Kelly of the Redstone Scientific Information Center for her tireless efforts in obtaining copies of so many references from distant libraries, Mr. Wayne Jones and Mrs. Mary Claire Woodriff for their efforts in computer coding, and Mr. John Chambers and Mrs. T. H. Ward for their editorial assistance.

TABLE OF CONTENTS

Chapter | Page

LIST OF TABLES

Page

LIST OF FIGURES

CHAPTER I. INTRODUCTION

This chapter introduces the report with a statement of the problem and discussion of the need followed by the scope of the research effort.

A. Statement of the Problem

The problem addressed in this report is to develop and demonstrate a methodology, with its associated computer model, that will acceptably transform several individual multicriteria rank-ordered lists of research and development (R & D) projects into a single, aggregated, *prioritized rank-ordered list to guide the investment* of R & D resources. In addition, provisions are needed for the individual lists to be converted from various formats. Decision-maker and judge self-rating weighting methodologies are necessary. The methodology developed must be capable of aggregating, with reasonably small effort, very long individual partial length and/or full length lists. Over fifty alternatives should be allowed in the full length lists.

B. Need for this Research Solution

The task of R & D management planning for high technology systems has become exceedingly difficult. The emphasis on coordination and communication between the management groups of major functional elements of high technology systems developmental organizations, both industrial and governmental, and the expanded usage of goal and objective planning methods have complicated the planning process. Situations have resulted where the planners in the R&D element receive many diverse

priority lists of suggested future R & D work or products from the other functional elements (i.e., marketing, field operations, production, and senior staffs) and from managers within the R & D organization. The prioritization criteria of interest to each contributing element differ as their functions and objectives differ. Therefore, the individual prioritizations are based to varying extent, upon the objective criteria and viewpoints of each element. The R & D element management must combine these lists into a usable list of prioritized R & D projects for their consideration in the allocation of discretionary R & D funds for use toward advancing the technology base of the enterprise.

As a specific example, the Director of the US Army Missile Laboratory has such a planning problem. Each year he selects and funds up to 100 R & D tasks with over $50 million. The Army policy of Single Program Element Funding (SPEF) gives to the Laboratory Director the final discretion to establish the most productive and best balanced program. The Director is provided an abundance of advice from outside and within his Laboratory. The advice is usually in the form of lists of R & D tasks or required products that should be given higher priority and, thereby, funding. To advise the Army Missile Laboratory Director, and his counterpart at other Army R & D Laboratories, the "user" staff agencies of the Army, which are the Deputy Chief of Staff for Operation (DCSOPS) and the Training and Doctrine Command (TRADOC), annually prepare documentations containing their priority rankings of potential, not yet developed, military materiel. The TRADOC list gives a rank-order number to each future system. The DCSOPS document, called the Science and Technology Guide (STOG) groups the potential materiel systems into unranked capability category classes of use; i.e., Air

Defense, Close Combat, Fire Support, etc. Within each capability category, the potential future systems are rank ordered. The Laboratory Directors must relate each R & D project technology task to the one or more potential future systems to which it could lead.

Three Headquarters organizations, the Department of Defense (DOD), the Department of the Army (DA), and the Development and Readiness Command (DARCOM), as well as the US Congressional staff all have, on occasion, sent letters to the Director recommending funding of certain selected tasks or groups of tasks within the Director's SPEF program. Tri-Service special topic committees prepare, and often prioritize, lists of R & D project tasks which they recommend for increased funding. Army Missile R & D technology areas of emphasis are included in the Tri-Service committee lists. The local Commander's staff also provides a rank-ordered list of systems to be supported by the Laboratory's R & D technology efforts. Often the Commander, who supervises the Laboratory Director, has a few R & D technology projects or potential applications that he believes should be given special attention and resources. Within the Army Missile Laboratory, task priority rankings are prepared by the Director's Staff and by subordinate managers of the Directorates and offices within the Laboratory. The Laboratory Director must allocate his discretionary funds to the R & D technology projects that will produce the most return for its investment cost to the Army and will maintain the viability of the Laboratory. He must give appropriate managerial, technical, and political weight to each of the recommendations of his advisors.

The preceding statement of need and the example resolve into a desirability for a methodology for aggregation of multiple criteria rank-ordered priorities.

C. Scope of this Research

The research documented by this report is based upon a comprehensive literature survey of the subjects of social choice and majority-rule methods that are applicable to the aggregation, without feedback, of multiple criteria rank-ordered ordinal priorities. From this basis, the research will determine and develop the specific majority-rule methodology to aggregate the variety of rank-ordered priority lists (as described previously) for R & D project priority determination. The chosen majority-rule methodology will be integrated into an aggregation logic model that will satisfy the following requirements:

1) Aggregate rank-ordered individual sublists which have any or all of the following features:

a) Complete length lists of up to 100 alternatives that rank all possible projects or requirements for products.

b) Reduced length lists (down to two alternatives) that rank less than the complete set of possible projects or requirements.

c) Transitive rank-ordered sublists.

d) Weak ordered and/or strongly ordered ($X \geq Y$ and/or $X > Y$) sublists.

e) Categorized grouping sublists, where one of the following may occur:

(1) The projects are subdivided into categories. Then, within each category, the projects are ranked, but the categories may or may not be strongly ranked.

(2) The projects are subdivided into categories. The categories are ranked, but the projects within a category may or may not be strongly ranked.

f) Multiple sublists ranked in accordance with a common criteria or with individual criteria.

g) Sublists where the alternatives are ordinal-ranked based upon various forms of cardinal utilities such as:

(1) Value estimates, or,

(2) The date that the usable materiel product from the R & D will be available.

2) Weight the importance and authenticity of each sublist during the aggregation process. The types of weighting mechanisms include the following:

a) Decision-maker weighting mechanisms to be applied to single alternative and/or to the sublists ranked by certain judges. The mechanisms will include multiplicative factors and exponential factors.

b) Judge self-expertise weighting where each judge will rate his own expertise on each alternative.

3) Analytically measure and statistically test the concordance of the set of sublist rank orders and the consistency of the aggregate rank order.

The logic model was converted into a digital computer code to perform the preceding requirements for up to 100 rank-ordered sublists.

The final logic model and computer code will be demonstrated in an R & D project prioritization study (see Chapter VII) tailored to resolve the Army Missile Laboratory Director's concerns as described in Section B of this chapter.

CHAPTER II. SURVEY OF PAST RELATED LITERATURE

This chapter contains the results of a comprehensive survey of the literature related to the research topic. The chapter contains a chronological and network analysis of the literature thrusts as well as synopses of key material. An extensive annotated bibliography was prepared and published in another report, Dobbins [13].*

A. Overview of the Literature

Extensive available literature was surveyed to select and evaluate the "Social Choice" relevant body of knowledge and its application to the problem of the aggregation of several ordinal R & D project preference rank orders into a single rank-ordered list. This final rank order should provide a reasonable representation of the consensus of the individual preferences. Since the known literature was primarily in the fields of welfare economics, political science, and social science, it had to be interpreted and translated to determine its relevance to R & D engineering management. For a better understanding of the material and for the application of the available knowledge, the relevant literature was separated into distinct thrust areas of emphasis. These thrust areas then were interrelated on a time axis to determine when and where the later thrusts branched off from the initial endeavors.

---

*[ ] denotes the number of a book or periodical cited in the references.

The search of the extensive relevant literature did not identify any work dealing directly with the goal of this report. There were no majority-rule type methods for aggregation of multiple criteria, ordinal rank-ordered lists to obtain a single rank-order list of R & D projects for resource investment. A few articles did recognize that R & D project selection was a possible application for social choice theory.

Certain restrictions were used to select the relevant material from the extensive body of social choice information. These restrictions include the following:

1) The rank orders of interest were ordinal, not cardinal, preferences.

2) The emphasis was on the aggregation of individual ranked preferences into single group rank-order lists.

3) The rank ordering and aggregation were single cycle, not temporal, decisions.

4) There was no feedback from individuals to other group members.

5) The individuals who ranked applied their sincere beliefs and were not using strategy to coerce the group result to agree with their preferences or objectives.

The literature selected was grouped into 25 thrust area groups of interest and of manageable size. The reference material was chronologically sequenced within each thrust area. To obtain an indication of the time span of work in each area, the dates of the earliest, the median, and the latest paper in each group were recorded in Table 1. Also listed is the number of papers and books relevant to this report that were grouped in each area. The first area, a) Pre-Arrow Basic, had four works that span from 1953 translations of the 1770

papers to 1958 with a median paper dated 1953. The last area, y) Fuzzy Set Rank Ordering, had ten papers which span from 1974 to 1979 with a 1978 median data. Except for two areas, Pre-Arrow Basic and Tullock's Books, all of the work areas appeared to be active at the writing of this report and more material can be reasonably expected to be published.

Figure 1 presents a network visualization of the interrelation along a time axis, of the 25 literature thrust areas. More specialized areas branch off from d) Majority-Rule Models thrust. The o) Tullock's Book's activity starts and ends within the time scale of this network. The network began to branch rapidly during the period immediately after the publications of Arrow's Impossibility Theorem. The current continuation of work in 23 of the 25 areas and the relatively late median dates listed in Table 1, reflect the accelerating activities of the 1970's, possibly kindled by the several excellent books published since the mid-1960's by authors such as Sen and Fishburn.

B. Synopses of Literature

The following paragraphs will give a brief synopsis of over 200 books and articles selected as relevant to the research for this report. Each book and article is summarized in somewhat more detail in Dobbins [13]. All are listed in the References. The material presented in each area of the synopses and Dobbins [13] has been discussed chronologically according to the publication year. The synopses are as follows:

1) Pre-Arrow Basic: Methods to use the preferences of the majority of voters to obtain a single consensus apparently were recorded [101] first in 1770 when Borda recommended his "method of marks."

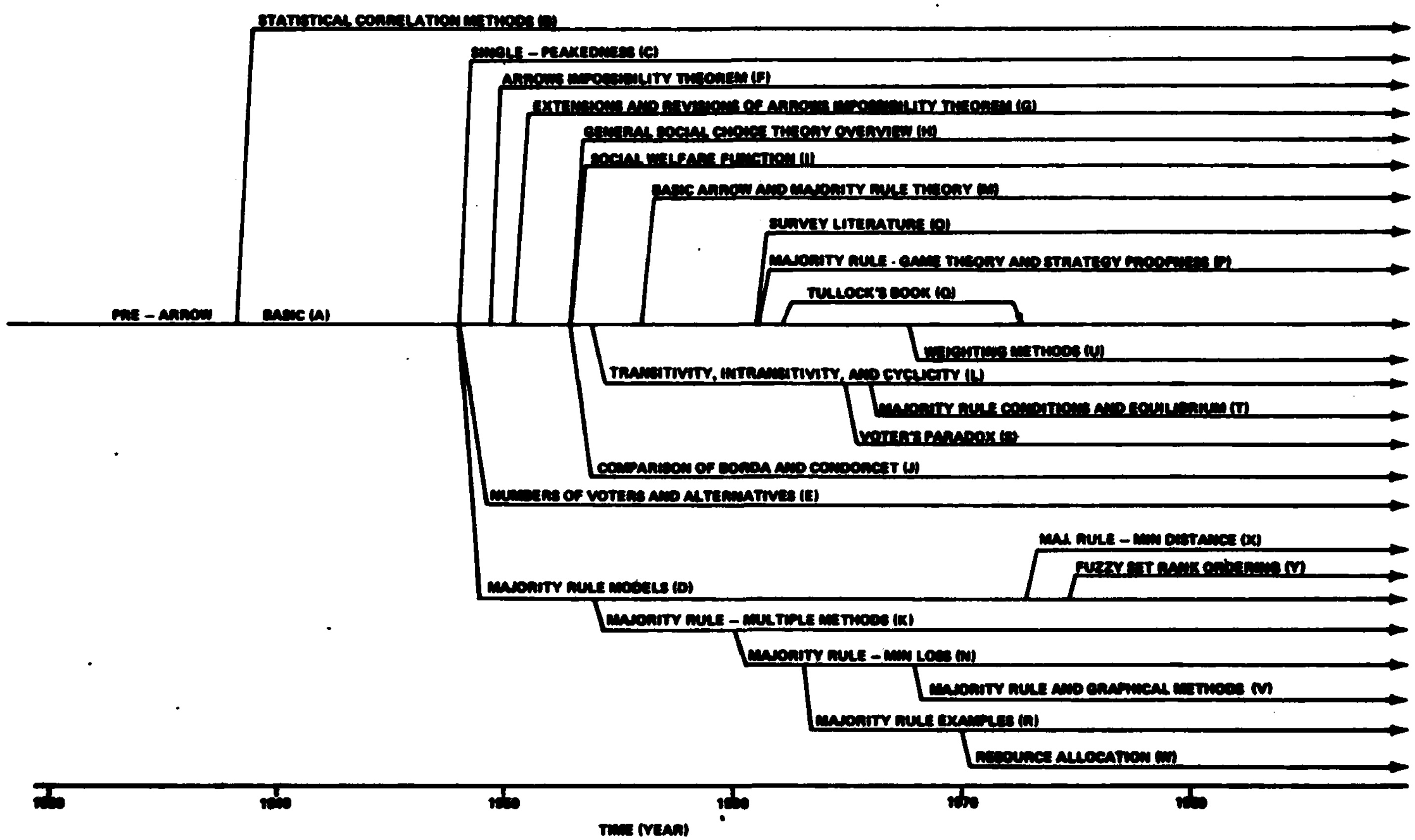


Figure 1. Individual preference aggregation literature growth tree.

TABLE 1. LITERATURE TOPIC GROUPS

Categories	Publication Date			No.
	First	Median	Last	
A. Pre-Arrow Basic	(1770)	1953	1958	4
B. Statistical Correlation Methods	1939	1956	1977	11
C. Single-Peakedness	1948	1975	1976	7
D. Majority Rule Models	1948	1973	1978	18
E. Number of Voters and Alternatives	1948	1976	1979	7
F. Arrows Impossibility Theorem	1951	1972	1978	14
G. Extensions and Revisions of Arrow's Impossibility Theorem	1952	1972	1978	23
H. General Social Choice Theory Overviews	1953	1973	1978	9
I. Social Welfare Function	1953	1974	1979	9
J. Comparisons of Borda and Condorcet	1953	1975	1977	20
K. Majority Rule-Multiple Methods	1954	1971	1978	8
L. Transitivity, Intransitivity, and Cyclicity	1954	1972	1978	19
M. Basic Arrow and Majority Rule Theory	1956	1974	1978	11
N. Majority Rule-Min Loss Methods	1960	1963	1976	4
O. Survey Literature	1961	1973	1978	15
P. Majority Rule - Game Theory and Strategy Proofness	1961	1975	1979	9
Q. Tullock's Books	1962	1969	1973	6
R. Majority Rule Examples	1963	1970	1975	3
S. Voter's Paradox	1965	1968	1976	7

TABLE 1. (CONCLUDED)

Categories	Publication Date			No.
	First	Median	Last	
T. Majority Rule Conditions and Equilibrium	1966	1973	1979	15
U. Weighting Methods	1968	1972	1975	6
V. Majority Rule - Graphical Methods	1968	1974	1976	4
W. Resource Allocation	1970	1971	1975	3
X. Majority Rule - Minimum Distance	1973	1974	1978	3
Y. Fuzzy Set Rank Ordering	1974	1978	1979	10

Condorcet's true majority and LaPlace's method followed closely after Borda's work. Dodgson (Lewis Carroll) recommended modifications to Borda's method during the mid-1800's. The economic interest in rank-ordered aggregation was kindled through Robbins' [31] 1932 contention, supported further by Bergson [58], that the magnitudes of individual preferences cannot be added but must be analyzed by ordinal means.

2) Statistical Correlation Methods: Kendall, Smith, Friedman, and Moran [22, 132, 161, 162, 163, 183] developed the statistical rank-ordered correlations, such as the coefficient of concordance method. This work has been further covered by others [15, 16, 19, 24, 26, 106].

3) Single-Peakedness: Black's contention [4, 62, 63] that single-peaked orders had preferred aggregation characteristics was substantiated and expanded upon by others [6, 23, 74, 103, 104].

4) Majority-Rule Methods: Various majority methods have been developed either as derivatives of the classical methods (Borda, Condorcet, etc.) or as new approaches. These include a vote score assignment equation by Schuler [210], dichotomous voting by Bartoszynski [56], pattern classification by Blin and Whinston [73], an aspiration level overlay by Harnett [146], a majority-rule Kendall-derived two-matrix method by Shannon [217], an extension beyond Shannon by Navarrete, et al. [185], a point system by Smith [221], a branch and bound algorithm method by Armstrong and Cook [50], others [88, 95, 126, 143, 220, 225, 227], and a conversion from ordinal to cardinal rank orders, before aggregation, method by Wood and Wilson [237].

5) Number of Voters and Alternatives: Fishburn [120, 124, 131] and Bell [57] developed several papers that quantify the likelihood

that a simple majority method will produce a winner as the quantity of voters and alternatives vary. Black, [64], Dutta and Pattaniak [105], and Greenberg [141] added theoretical depth to these data.

6) Arrow's Impossibility Theorem: K. Arrow, in 1951, determined [1, 51] the required conditions for a social welfare function which aggregated individual preferences, then proved the theorem that a fair social welfare function, without a dictatorship, was impossible. The required conditions were more simply and clearly stated by Little [169] as follows:

a) Retrievability of alternatives.

b) An alternative's relative position in an individual order will not relatively change in the aggregate order.

c) The independence of irrelevant alternatives.

d) Non-imposition.

e) Non-dictatorship.

Numerous researchers countered his theorem [82, 189], offered modifications [99, 108, 181], or presented alternative proofs [111, 173, 179, 211] of Arrow's results.

7) Extensions and Revisions of Arrow's Impossibility Theorem: While striving to overcome the "impossible," social science researchers developed new or modified conditions for social welfare functions that could be satisfied, albeit from a reduced variety of acceptable individual orders. May [171] defined conditions of decisiveness, symmetry, neutrality, and positive responsiveness. Other conditions defined are summation of ranks [94], unanimity and monotonicity [67], split groups of indifferent alternatives [152], intensity added to the independence of irrelevant alternatives [86], intensity of antagonism index [168],

anonymity [110], group rationality [109]. Others reaffirmed Arrow's Impossibility results through different assumptions [53, 66, 75, 97, 115, 144, 145, 148, 151, 191, 213, 214, 233, 234, 235].

8) General Social Choice Theory Overview: As knowledge developed, several comprehensive social choice theory works became available from authors such as Luce and Raiffa [27], Pattanaik [28, 29], Fishburn [17], Blin [69], Herzburger [147], Ferejohn and Page [112], and Aumann [3].

9) Social Welfare Function: The definitions and conditions for social welfare functions have been analyzed [2, 102, 114, 116, 135, 160, 193, 239], while the question of the existence of such functions has been debated [158, 205, 208].

10) Comparison of Borda and Condorcet: In recent years many researchers have comparatively analyzed the majority-rule methods developed by Borda and Condorcet. De Grazia [101] and Black [4] provided translations and analyses of the original papers. Fishburn [119, 129], Young [238, 240], and Richelson [200, 202, 203] compared the two methods. The Condorcet method was the reference method of several papers [129, 137, 190], while Borda's method was used for others [49, 77, 94, 116, 121, 125, 130, 180, 241].

11) Majority Rule-Multiple Methods Compared: Goodman described [139] the Copeland two-step method and compared it favorably to other methods, as did Richelson [200, 202, 203]. Svestka [40], Wyatt [47], Chartier and Wertheimer [91], and Castore, Peterson, and Goodrich [90] made comprehensive comparisons of lists of majority rule and other aggregation methods.

12) Transitivity, Intransitivity, and Cyclicity: An accepted limitation of the unrestricted use of many majority-rule methods is

the real probability of aggregated results which are intransitive or cyclic. Many researchers have worked to better understand the characteristics and causes of transitivity, intransitivity, and cyclicity, and to develop conditional restrictions that will control these aspects of rank-ordered aggregation. May [172] and others strived to clarify intransitivity [117, 156, 174, 212], while Inada and others [76, 98, 100, 118, 140, 153, 155, 157, 167, 178, 198, 209, 215] developed decision rules to avoid intransitive results, generally through restrictions on the characteristics of the individual rank orders. Pomeranz and Weil [196] computed probabilities of cyclical majorities.

13) Basic Arrow and Majority Rule Theory: New axiomatic structures for preference theory were developed by Luce [170], Fishburn [114, 122, 123, 127, 128, 140], and others [93], while Koopman [166] clarified concepts in existing theory.

14) Majority Rule-Minimum Loss Methods: Van den Bogaard and Versluis [44] and others [21, 150, 227] developed techniques to aggregate individual welfare preferences by minimization of the social loss function which was based upon individual loss functions.

15) Survey Literature: Surveys of the work and publishings of others in the field were made by Riker [in Coombs 10, 199], Guilbaud in Lazerfield and Henry [25], Sen [34], Fishburn [17], Plott [195], and others [6, 7, 8, 11, 30, 32, 33, 39, 42, 81].

16) Majority Rule-Game Theory and Strategy Proofness: Based upon the 1947 work of von Neumann and Morgenstern [45], game theory had been applied to the simple majority concept. Barbut [54] introduced the relation of the two-person, zero-sum game to majority-rule methods, while Shisko [36] extended this approach for n-person majority-rule

games. The relevant theory was developed and extended [84, 85, 175, 176, 184, 206, 207]. Gibbard [136] applied the game theory approach to develop the criteria for a strategy-proof voting scheme.

17) Tullock's Books: G. Tullock stated in his books [5, 43, 228] that Arrow's Impossibility Theorem would seldom be important and that little had been contributed in the majority-rule area since Black developed single-peakedness. Tullock's critical writings generated responses from Arrow [52] and others [20, 55].

18) Majority Rule Examples: A few reports were found that represented practical examples of the application of the majority rule [87, 218, 229].

19) Voter's Paradox: Effort has continued to generalize the analysis to determine the probability of achieving an intransitive aggregate rank-order through a majority-rule method [48, 89, 133, 134, 164, 186, 231].

20) Majority-Rule Conditions and Equilibrium-Diverse efforts have continued striving to determine the conditions required for a usable majority-rule decision. Murakami [182] determined that the scoring constants should logically be 1, 0, -1, have autonomy, be non-reversed, and have non-dictatorship. Pattanaik [192] pursued the value restrictions techniques, while Fine [113] recommended monotonicity and faithfulness. Blau and Deb [68] proved that an infinite social decision function does not produce a choice. Richelson [201] classified and showed the interrelations of the multitude of overlapping conditions for a social choice function. Others [35, 38, 149, 159, 177, 187, 197, 216, 219] also worked in this area.

21) Weighting Methods: Weighting permits special importance to be considered for certain judges' rankings or for certain alternatives. Winkler [236] and others [92, 165] summarized the most useful methods: equal weights, weights proportional to ranking, weights proportional to self-rating, and weights based on previous assessments. Gustafson, Pai, and Kramer [142] reported on a weighting method for hierarachical R & D project selection. Rowse, Gustafson, and Ludke [204] added peer weights, group weights, and average weights to the list of available methods. Einhorn and Hogarth [107] discussed unit weighting.

22) Majority Rule-Graphical Methods: Research has continued toward describing the social decision process through graphical structures [21, 46, 138, 224, 232].

23) Resource Allocation by Voting: Several studies were made of the effect of voting rules on resource allocation processes such as capital budgeting and R & D project selection [12, 18, 61, 80, 194, 222].

24) Majority Rule-Minimum Distance Technique: Bowman and Colantoni [78, 79] developed a majority-rule rank-ordered decision method based upon the minimizing of a decision function defined as a one-dimensional "distance." The scoring constants used are 1, 1/2, 0 and the distance function is $d\ (p^*, \overline{P}) = f\ [\Sigma_{ij} d_{ij} (P^*_{ij}, \overline{P}_{ij})]$. Blin and Whinston [72] suggest a quadratic assignment solution to the Bowman and Colantoni method. Cook and Seiford [96] extended the one-dimensional distance minimization method so that it can be solved by linear programming techniques.

25) Fuzzy Set Rank Ordering: Blin originally proposed [70, 71] that fuzzy binary relations could be used for group preference orderings.

Others [154, 188, 223, 230, 233] especially Bezdek, Spillman, and Spillman [59, 60] have developed Blin's proposition into a new area of social choice study. Buckles [83], as class work, translated from Fuzzy Set terminology to that of majority rule.

CHAPTER III. PRIORITIZED RANK ORDER AGGREGATION METHOD CONCEPT

In this chapter, the concepts of ordinal aggregation are presented by first introducing the foundation considerations, then presenting a comparative analysis of majority-rule methods.

A. Foundation Considerations

1. Need

The majority-rule concept chosen for the objective method of this research must be flexible in its application as well as technically satisfactory. It must accept complete and partial orders, strong and weak preferences, and weighting factors. Also, it must not be negated by intransitivities. The quality of the whole aggregated rank order is more significant than the precision of the first choice winner selection. For resource allocation purposes where, for example, the first 40 projects are ranked, the identity of the number one project is of passing interest, while being in the first segment (say, the first 10%) can be of pressing importance.

2. Known Method Characteristics

Various aggregation rank-ordered generating methods relevant to this research have been surveyed and seven have been chosen for comparative analysis. None of the methods chosen utilized feedback of group consensus to the judges nor dialogue between judges. Further, none of the chosen methods required cardinal utilities to measure preferences. These restrictions also eliminated the Delphi method and

the unfolding method offered by Coombs [10]. The family of methods considered further was variations of the majority-rule method. For problems with only two alternatives these methods all resolved to the same simple majority-rule method where the winner was the alternative that had the majority of the votes. But for three or more alternatives, several defined methods were available. These majority-rule methods are as follows:

a) Borda's Method of Marks.

b) Adjusted Borda Method of Marks.

c) Condorcet's Criterion.

d) Black's Simple Majority Procedure.

e) Dodgson's Method of Inversion.

f) Copeland's Majority Rule.

g) Shannon and Svestka's Majority Rule Method.

h) Black's Single-Peaked Preference.

Further examination of the theory and practice of h) Black's Single-Peaked Preference determined that it was not a practical method for aggregation with a large number of alternatives because the only known method required consideration of all alternative combinations as possible reference orders. After an explanatory description in Section 4 of this chapter, this method will not be evaluated.

Majority-rule methods were recommended by Kenneth Arrow as the better method for social choice aggregation. Arrow's book [1] is credited by Blau [67] with presenting the first organization of the social choice problem as a deductive system. Since interpersonal comparison of utility had been excluded, the theory of ordinal relations became dominant for social ordering. Arrow's work showed that when the

axioms (conditions) of economic social choice were fully defined, they were inconsistent. Arrow defined the Social Welfare Function (SWF) as a technique to express the aggregate economic preferences of society. Arrow's conditions for SWF were the foundation of the current social welfare theory, but they were considered by many writers as too restrictive. His conditions were highly technical and difficult to understand. Therefore, each Arrow condition will be replaced by a more easily understood interpretation by Svestka [40].

Condition 1:

A social welfare function (amalgamation method) must define a unique order, given any subset of all possible orders.

Condition 2:

If one alternative rises or remains stationary in the order of every individual, then it must not fall in the joint order.

Condition 3:

The removal or insertion of an alternative in the set of alternatives which result in no change in any individual order of the remaining alternatives must not cause a change in the order of the remaining alternatives of the joint order.

Condition 4:

The joint order is a function of the individual orders and must not be imposed by some outside influence.

Condition 5:

The joint order must not be arbitrarily defined by the order of one individual without consideration of the other individual orders.

Many theoretical writers responded to Arrow's conditions published in 1951. Based on certain of these critiques, Arrow revised three of

his conditions. Conditions 1, 2, and 5 were strengthened, but the condition inconsistency represented by Arrow's Impossibility Theorem still held. Arrow further wrote that for more than two alternatives the method of majority (majority rule) satisfied Conditions 2 through 5 and a restricted form of Condition 1 (labeled 1'), and thus was like a social decision function, even if not an SWF. Arrow did not specify precisely a majority-rule method for more than two alternatives. Promising methods from the literature will be described later in this chapter. To aid in understanding the material to follow, certain social choice notation will be explained:

xPy - means that the alternative x is strictly preferred to the alternative y ($x > y$).

xMy - means that in paired comparative voting, alternative x has a majority of the votes over alternative y.

xRy - means that alternative x is equal to or preferred to alternative y ($x \geq y$).

xIy - means that the judge is indifferent between alternative x and alternative y ($x = y$).

3. Comparison Characteristics

The seven majority-rule methods will be compared to determine the one that is preferred for the aggregation model for this research. They will be compared, similar to Wyatt [47] and Richelson [200, 202, 203], by the requirements that must be met by the individual sublist orders before the aggregation method can be utilized, and by conditions of majority-rule performance. The requirements and procedural conditions considered for the methods are as follows:

a) Only Cardinal Utilities Required? - Must the method accept only cardinal rank order data, or are ordinal data accepted?

b) Only Transitive Orders Required? - Can the method also accept intransitive rank orders?

c) Only Complete Orders Required? - Can the method also accept partial orders?

d) Only Strong Orders Required? - Can the method also accept weak rank orders?

e) Restricted Number of Judges or Alternatives? - Is the method restricted to an odd or even number of judges or alternatives, for example?

f) Not Have Condorcet Extension Procedure? - If an alternative has a simple majority over every other alternative, will it be the first place choice? (See Richelson [200]).

g) Not Have Cancellation Property? - Will any yPx in an individual's preference ordering be balanced by any xPy in some other individual's ordering as long as there are no other alternatives between x and y in either ordering? (See Richelson [200]).

h) Difficult to Enlarge? - Is the implementation of the method significantly more difficult as the number of judges or number of alternatives increases from three?

i) Produces Only Winners? - Is the method intended to produce full aggregated rank orders or only the first place alternative?

4. Descriptions of Compared Methods

The following is a more detailed discussion of the majority-rule methods compared in this dissertation.

a. Borda's Method of Marks

Black [4, 65] interpreted Jean-Charles de Borda's paper, printed in 1781 in the Memoirs of the French Academy of Science, to offer two methods which gave the same results. The second of Borda's stated methods gave each alternative a mark equal to the sum of the votes that it would get when it was put against each of the other alternatives, individually. When the alternatives were x, y, and z, if y were placed first in the preferences of any given judge, it would defeat x and z and would secure two votes. If x stood second in the judge's preference, it would defeat one alternative and secure one vote. This method had been demonstrated by Black to hold for strong orders.

Fishburn [119] and Richelson [200] both defined Borda's social decision function as:

$$f(Y,D) = \left\{x : x \epsilon Y \text{ and } \sum_{i=1}^{n} r_i(x,Y,D) \geq \sum_{i=1}^{n} r_i(y,Y,D) \text{ for all } y \epsilon Y\right\}$$

where r_i is the rank assigned to each alternative by judge i, Y is the set of feasible alternatives, and D is the n-tuple of individual linear orderings on Y. For each alternative, the Borda score was the sum of the marks it received from the judges, where, for m alternatives, the marks given were, in decreasing order of preference, (m-1), (m-2) ..., 0. The ranking was based on the scores of each alternative; i.e., the highest score won.

Example 1:

An example of the Borda method, with N = 5 judges and M = 4 alternatives is:

Judge 1: $w > x > y > z$

Judge 2: $x > y > z > w$

Judge 3: $w > x > y > z$

Judge 4: w > x > z > y

Judge 5: x > z > y > w .

By competing each alternative against all other alternatives, the following matrix of votes is obtained:

	w	x	y	z	Borda Row Totals
w	-	3	3	3	9
x	2	-	5	5	12
y	2	0	-	3	5
z	2	0	2	-	4

In this matrix, for example, when x competed against w, there were 3 votes for w over x and 2 votes for x over w. The Borda winner for the example problem is x with 12 votes total and w is second with 9. The Borda rank order is x > w > y > z.

b. *Adjusted Borda's Method of Marks*

Black [4, 65] further interpreted Borda's work to cover orders where some judges were indifferent between certain alternatives. Black explained the Adjusted Borda method as that of assigning marks on the basis of one mark for each alternative it stood above and deducting one mark for each alternative it stood below. In terms of the matrix analysis, Black said that this method assigned to an alternate a plus mark equal to the total number of alternatives above its place on all of the sublists, and a minus mark equal to the total of the alternatives below its place. The plus portion could be calculated by summing the figures in the row cells for that alternative and the minus portion was obtained from summing the column cells for that alternative.

Fishburn [119] defined the Adjusted Borda social decision function as

$$F(Y,D) = \left\{x : x \varepsilon Y \text{ and } \sum_{i=1}^{n} S(x,Y,P_i) \geq \sum_{i=1}^{n} S(y,Y,P_i) \text{ for all } y \varepsilon Y\right\}$$

where

$$s(x,Y,P_i) = \left|\left\{y : y \varepsilon Y \text{ and } xP_i y\right\}\right| - \left|\left\{y : y \varepsilon Y \text{ and } yP_i x\right\}\right| .$$

Again the ranking was based on the net scores for each alternative with the most positive score winning.

Example 2:

An example of the Adjusted Borda Method with N = 5 judges and M = 4 alternatives is:

Judge 1: w = x > y = z

Judge 2: y > z = w > x

Judge 3: x > y = w > z

Judge 4: w = z > x > y

Judge 5: y > w = x = z

which resolves to the following vote matrix; if indifferences between pairs are each given one-half vote:

	w	x	y	z	Borda Row Totals	Alternate Borda Totals
w		3	2 1/2	3 1/2	9	3
x	2	-	3	2 1/2	7 1/2	0
y	2 1/2	2	-	3 1/2	8	1
z	1 1/2	2 1/2	1 1/2	-	5 1/2	-4
Borda Column Totals	6	7 1/2	7	9 1/2		

For the one-half vote to score, the Borda method of marks technique had the Borda Row Totals Column with the Borda marks; i.e., 9, 8, 7 1/2, 5 1/2 with order w > y > x > z. But with the Adjusted Borda Method, the total scores which were the row scores minus the column scores, were 3, 1, 0, -4 with order: w > y > x > z. The Borda and Adjusted Borda Methods, using one half votes for ties, gave the same rank orders.

But if the ties were calculated as Black [4] suggested that Borda prescribed them, then last ranked tied alternatives in a judge list were given zero. Those above the lowest rank, and tied, were both given the votes to represent the position of the tied pair. Using the classical method for the Borda and Adjusted Borda counts, Example 2 will give the following:

	w	x	y	z	Borda Row Totals	Adjusted Borda Totals
w	-	2	2	2	6	3
x	1	-	3	2	6	0
y	2	2	-	3	7	1
z	0	2	1	-	3	-4
Borda Column Totals	3	6	6	7		

By the original Borda tie counting method, the Borda count is 7, 6, 6, 3 and the Adjusted Borda count is 3, 1, 0, -4. The corresponding rank orders are Borda: y > w = x > z and Adjusted Borda: w > y > x > z. The Adjusted Borda methods for both tie scoring techniques gave the same rank order, but the Borda rank orders differed. Black [4] emphasized that the Borda method was inappropriate where indifference (ties) existed.

c. Condorcet's Criterion

Black [4], Fishburn [119], and Fishburn and Gehrelein [129] all interpreted the Marquis de Condorcet's essay, published in 1785 to say that, if, out of three or more alternatives, one was able to get a majority over each of the others, it ought to be selected. The implication was made that another method must be used if the one alternative does not have the Condorcet majority.

Fishburn [119] defined Condorcet's proposal as a decision function,

$F(Y,D) = C(Y,D)$ whenever $C(Y,D)$ existed

where

$C(Y,D) = \{X : X \in Y \text{ and } xMy \text{ for all } y \in Y - (x)\}$;

thus, either

$C(Y,D) = 1$ or ϕ, where ϕ meant a Condorcet winner did not exist. This also proposed that if an alternative had a simple majority over each other alternative, then it should win. The Condorcet *criterion* picked a winner. The second place might be picked by the same rule after the first place alternative had been selected and its data withdrawn. There was, in each aggregation, a strong probability of no winner by the Condorcet criterion.

In Example 1 for the Borda Rule Method, the Condorcet evaluation of the majority winner of pairs, in matrix form, would be as in the example that follows, which will be labeled Example 3:

Example 1

	w	x	y	z
w	-	3	3	3
x	2	-	5	5
y	2	0	-	3
z	2	0	2	-

Example 3

	w	x	y	z
w	-	1	1	1
x	0	-	1	1
y	0	0	-	1
z	0	0	0	-

In Example 3, a 1 in Row w, Column x (w,x) means that, in Example 1, wMx (w has a majority over x); a 1 in w,y means wMy and a 1 in w,z means wMz. In the second row a 0 in x,w means not xMw; therefore, w does have a majority over all other alternatives but x does not. W is the strong Condorcet winner. Neglecting w data, x is the Condorcet majority over all other candidates. The Condorcet rank order was w > x > y > z while the Borda rule rank order was x > w > y > z.

The same Condorcet procedure applied to Example 2 produced the following matrix as Example 4:

Example 2

	w	x	y	z
w	-	3	2 1/2	3 1/2
x	2	-	3	2 1/2
y	2 1/2	2	-	3 1/2
z	1 1/2	2 1/2	1 1/2	-

Example 4

	w	x	y	z
w	-	1	1/2	1
x	0	-	1	1/2
y	1/2	0	-	1
z	0	1/2	0	-

for w, wMx, wRy, and wMz, so w is a weak Condorcet winner. For x, the preferences are not xMw, but xMy and xRz. Therefore x is a weak Condorcet second place in the rank order. Examples also can be shown where there are no Condorcet winners.

d. Black's Simple Majority Procedure

Black [4] recommended a procedure where the Condorcet procedure was applied first to determine an alternative that had a majority over all other alternatives. If no Condorcet winner existed, Black's second step was to apply the Borda method, or, if indifferences existed, the Adjusted Borda method.

Richelson [200] defined this Black method as the social decision rule:

$$F(Y,D) = \{x : x\varepsilon Y \text{ and } xMy \text{ for all } y\varepsilon Y\}$$

$$\text{IFF } \{x : x\varepsilon Y \text{ and } xMy \text{ for all } y\varepsilon Y\} \neq 0$$

$$= \left\{ x : x\varepsilon Y \text{ and } \sum_{i=1}^{n} r_i(x,Y,D) \right\}$$

$$\geq \left\{ \sum_{i=1}^{n} r_i(y,Y,D) \text{ for all } y\varepsilon Y \right\} \quad \text{otherwise.}$$

e. Dodgson's Method of Inversion

Black [4] interpreted the 1873 to 1885 writings of the Rev. Charles Lutwidge Dodgson on committees and elections. Dodgson also wrote under the pseudonym Lewis Carroll. Dodgson was first to use matrix notation and to well understand acyclicity. His method was that if initially there was no majority alternative, the decision maker should indicate which alternate would win by virtue of the smallest amount of change in the rating rank orders of the judges.

Richelson [*200, 202, 203*] *defined* Dodgson's procedures as the decision rule:

$$F(Y,D) = \{x : x\varepsilon Y \text{ and } t(x,Y,D) \leq t(y,Y,D) \text{ for all } y\varepsilon Y\}$$

where $t(x,Y,D)$ is the least number of inversions needed for x to obtain a simple majority over every other alternative. An inversion occurred, for example, if one x, y in a preference ordering was changed to y, x.

Example 5: For example, if N = 5 judges and M = 4 alternatives, and

Judge 1: $w > x > y > z$

Judge 2: $x > y > z > w$

Judge 3: $y > w > x > z$

Judge 4: $w > x > z > y$

Judge 5: $x > z > y > w$

the matrices would be as follows with the left one giving vote sums and the right one showing pair majority results:

Example 5

	w	x	y	z
w	-	3	2	3
x	2	-	4	5
y	3	1	-	3
z	2	0	2	-

	w	x	y	z
w	-	1	0	1
x	0	-	1	1
y	1	0	-	1
z	0	0	0	-

Now by inverting only one pair in Judge 3's rank order from w > x to x > w, the sum for the pair would cause xMw and alternative x, which has a majority over all other alternatives, becomes the Dodgson winner. The second place Dodgson alternative would be w, with y as the third place. By a different single inversion, w could be the Dodgson winner.

f. Copeland's Majority Rule Method

Goodman in Thrall [41] reported the work of A. H. Copeland, presented as mimeographed notes titled "A 'Reasonable' Social Welfare Function," November 1951, at a University of Michigan Seminar on Applications of Mathematics to the Social Sciences. The method, which permitted row ordering, was paraphrased by Goodman as:

If sgn x be the signum function of x ("signum" is defined from Latin as "a sign" or "a physical representation"):

$$\text{sgn } x = \begin{cases} +1 \text{ for } x > 0 \\ \ \ 0 \text{ for } x = 0 \\ -1 \text{ for } x < 0 \end{cases}$$

(x > 0 is interpreted to mean that for an array of i, k cells the row term i is greater than the column term k and x < 0 would mean k > i).

Then,

$$\text{sgn } \hat{\phi} = \begin{cases} +1 \text{ for } i > k \\ \ \ 0 \text{ for } i = k \\ -1 \text{ for } k < i \end{cases}$$

where

$\hat{\phi}(i) = \sum_{k=1}^{s} \text{sgn}\ \hat{\phi}(i,k)$ is the score for row i. The row with the greatest $\phi(i)$ is chosen.

Fishburn [17, 119] defined the decision function as:

$F(Y,D) = \{x : x \in Y \text{ and } s(x,Y,D) \geq s(y,Y,D) \text{ for all } y \in Y\}$

where

$s(x,Y,D) = \#\{y : y \in Y \text{ and } xMy\} - \#\{y : y \in Y \text{ and } yMx\}$, which means that $s(x,Y,D)$ is the number of alternatives in Y that x had a strict simple majority over, minus the number of alternatives in Y that had a strict simple majority over x. Fishburn emphasized that this method was not intended for weak orders.

Richelson [200, 202, 203] defined the Copeland method function the same way as Fishburn did with xMy defined to be that x was preferred by a majority to y; i.e., $n(x,y) > n(y,x)$, which is a strong rank-order definition.

By applying the Copeland method to Example 1, Example 6 was obtained:

Example 1

	w	x	y	z
w	-	3	3	3
x	2	-	5	5
y	2	0	-	3
z	2	0	2	-

Example 6

	w	x	y	z	s Score
w	-	+1	+1	+1	+3
x	-1	-	+1	+1	+1
y	-1	-1	-	+1	-1
z	-1	-1	-1	-	-3

In the s score column are the Copeland scores for this aggregation: w:+3, x:+1, y:-1, and z:-3 which gives a rank order of w > x > y > z that is the same as the Condorcet rank order for Example 3.

By definition, the Copeland method should not be used for Example 2 which had weak rank orders.

Fishburn [17, 119] presented an example:

Judges 1, 2, 3, 4 $y > x > a > c > b$

Judges 5, 6, 7 $b > c > y > a > x$

Judges 8, 9 $x > a > b > c > y$

which has x as the Copeland method winner, y as the Borda method winner, and no Condorcet winner. This is shown as Example 7:

	x	y	a	b	c	Borda	Adj Borda		x	y	a	b	c	Copeland Score
x	-	2	6	6	6	20	+4	x	-	-1	+1	+1	+1	2
y	7	-	7	4	4	22	+8	y	+1	-	+1	-1	-1	0
a	3	2	-	6	6	17	-2	a	-1	-1	-	+1	+1	0
b	3	5	3	-	5	16	-4	b	-1	+1	-1	-	+1	0
c	3	5	3	4	-	15	-6	c	-1	+1	-1	-1	-	-2
Col Totals	16	14	19	20	21									

The Copeland method rank order is

$x > y = a = b > c$,

while the Borda methods rank order is

$y > x > a > b > c$.

Fishburn pointed out that x won the Copeland method over y even though in the x,y competition y had seven votes and x had only two.

g. Shannon and Svestka's Majority Rule Method

Svestka [40], Shannon [217], and Wyatt [47] described the Shannon and Svestka majority-rule method, which will be called the "Shannon Majority-Rule Method" throughout the remainder of this dissertation. Svestka described the method as, first, the aggregation of the individual orders through the use of the Kendall array. Next, the Kendall array was modified to form a second array. Since the sum of two opposing cells (x,y and y,x) in the first array equaled the number of judges voting for the alternatives represented by that cell

(x,y); that cell, of each pair, containing the larger number of votes dominated its opposing cell Therefore, the numbers in all dominating cells were replaced by one nd the numbers in all dominated cells by zero. The new row values were summed into the a_i column. The rank order was given by decreasing magnitudes of the a_i values. i.e., the most preferred had the largest a_i value.

Shannon [217] and Wyatt [47] described the method as first constructing matrices for each judge where each alternative was compared to the others and a one was placed in each row-column intersection where the row alternative was preferred over the column alternative. The major diagonal was disregarded. The judge frequency matrices were added to form a summed frequency matrix where a count was made for each alternative of the number of times it was preferred to each of the other alternatives. In case of indifference (ties by a judge) between two alternatives, the single vote was split with one-half vote to each alternative. The sum of the complement cells, in the summed frequency matrix, even with ties, equaled the number of judges voting on that pair of alternatives. Next, the second matrix, the preference matrix, was developed by a comparison of the number of votes in the complement cells (x,y versus y,x). If the x,y value was greater than the y,x value, then x dominated y and a 1 was placed in the x,y cell and a zero in the y,x cell of the preference matrix. If the x,y value equaled y,x, then each complement cell in the matrix was given one-half. The cells in each row of the preference matrix were summed. The alternatives were ranked by the order of their preference matrix row sums, a_i. The row sum figures, a_i, were the number of times the row alternatives had a majority of votes over each of the other alternatives.

In terminology like Copeland's, the Shannon method can be defined, if sgn is the signum function, as

$$\text{sgn}\ \hat{\phi} = \begin{cases} +1 & \text{for } i > k \\ 1/2 & \text{for } i = k \\ 0 & \text{for } k > i \end{cases}$$

where

$$\hat{\phi}(i) = \sum_{k=1}^{r} \text{sgn}\ \hat{\phi}(i,k)$$

is the score of Row i. The rows were ranked based on their $\hat{\phi}(i)$ magnitudes. In Fishburn's and Richelson's terminology, the Shannon method would be defined as

$$F(Y,D) = \{x : x \varepsilon Y \text{ and } s(x,Y,D) \geq s(y,Y,D) \text{ for all } y \varepsilon Y\}$$

where

$$s(x,Y,D) = \#\{y : y \varepsilon Y \text{ and } xRy\}$$

which means that

s(x,Y,D) is the number of alternatives in Y over which x had a weak simple majority.

By applying the Shannon method to Example 1, Example 8 was obtained:

	Sum Frequency Matrix Example 1					Preference Matrix Example 8				
	w	x	y	z		w	x	y	z	s Scores
w	-	3	3	3	w	-	1	1	1	3
x	2	-	5	5	x	0	-	1	1	2
y	2	0	-	3	y	0	0	-	1	1
z	2	0	2	-	z	0	0	0	-	0

The Shannon a_i scores are in the s column: w:3, x:2, y:1, z:0 which gives a rank order of

w > x > y > z

which is the same as the Condorcet and the Copeland rank order.

But unlike the strict order Copeland method, the Shannon method can be used for Example 2 and will give the same preference matrix as the matrix in Example 4 for the Condorcet rank order winner. However, the Shannon Method will give a rank order for an example where the Condorcet method might only give a first place winner for that example.

h. Black's Single-Peaked Preference

Black [4] presented a theorem on single-peaked preference curves of judges to say that a simple majority would be obtained by an alternative if all judges' preference curves were single-peaked and there was an odd number of judges. Black said that this theorem applied for any majority-rule procedure. In definition, Black said it would be possible, when preference orders were represented by two-dimensional diagrams, to find an ordering of the points on the horizontal axis that left the curve as one which changed its direction at most once from up or down. The key seemed to be finding an ordering for the horizontal axis. This researcher has not found where Black nor other researchers have developed procedures to simplify the search for this one order that will permit single-peakedness. This combinatorial problem grows progressively worse as the number of alternatives increases above three.

B. Majority-Rule-Methods Comparisons

1. Majority-Rule Methods

The seven majority-rule methods were compared to determine if they satisfied the classification categories that were discussed in Chapter III. The classification categories were worded for yes or no answers, with a no answer preferred for the purposes of this research. Table 2 presents the results of the comparison of methods. None of

TABLE 2. MAJORITY RULE METHODS CLASSIFICATION BY REQUIREMENTS AND PROCEDURES

	Requirements Characteristics			Procedural Characteristics				
Methods	Only Transitive Orders	Only Complete Orders	Only Strong Orders	Restricted No. Judges No. Alternatives	No Condorcet Extension	No Cancellation Property	Difficult To Enlarge	Produces Only Winners
a) Borda's Method of Marks	No	No	Yes	No	Yes	No	No	No
b) Adjusted Borda's Method of Marks	No	No	No	No	Yes	No	No	No
c) Condorcet's Criterion	No	No	No	No	No	No	No	Yes
d) Black's Simple Majority Procedure	No	No	No	No	No	No	Mod	No
e) Dodgson's Method of Inversion	Yes	Yes	Yes	No	No	Yes	Yes	No
f) Copeland's Majority-Rule Method	No	No	Yes	No	No	No	No	No
g) Shannon and Svestka's Majority-Rule	No	No	No	No	No	No	No	No

the methods required only cardinal utilities; therefore this classification category was not tabulated.

Four methods have only one yes or MOD rating. They are: b) Adjusted Borda; c), Condorcet; d), Black's Simple Majority; and f), Copeland. The b), Adjusted Borda method decides on marks, not majorities; c) Condorcet does not necessarily have a winner nor a rank order. The d) Black's Simple Majority method, a combination of c) Condorcet and b) Adjusted Borda or a) Borda, is rated as a "moderately" difficult method to enlarge for numerous judges and numerous alternatives and has either the yes of Condorcet or Adjusted Borda, or the two yes's of Borda.

Only one method does not require a yes rating. It is g) the Shannon and Svestka's majority-rule method. Because of these ratings and the method's flexibility and adaptability, this method has been chosen as the main thrust of the aggregation model. Three other promising methods, the two Borda's and Copeland's, will be investigated further during the research.

2. Aggregation Concept Selected

The Shannon and Svestka Majority-Rule Method, described in Section A4 g of this chapter, was chosen as the basic aggregation method for this research. An appealing advantage of this method is that cyclic or intransitive type aggregate rank orders appear only as indifferent alternatives.

Example 8, Section A4 g, was an abbreviated example of the method. Example 9 is a more detailed description of the Shannon Method. Let the five judges' rankings of four alternatives (w, x, y, z) be as follows:

Example 9

Judge 1: $w > x = y > z$

Judge 2: $x > z = y$

Judge 3: $y > w = z > x$

Judge 4: $w > y > z > x$

Judge 5: $z > y$

For each judge, a frequency matrix was constructed to show when each alternative was preferred over each other alternate. A judge was alloted one vote per pair of alternatives; therefore, a tied pair gave one-half vote to each alternative in the pair. The alternative for each row was compared to all column alternatives and a 1 was placed at each intersection where the row alternative was preferred to the column alternative. Disregarding the major diagonal which is the intersection of like alternatives, Judges 1, 2, and 3 voted indifferently toward several alternatives. Judges 2 and 5 did not include all four alternatives in their partial rank orders. The individual frequency matrices for Example 9 are as follows:

FREQUENCY MATRICES (EXAMPLE 9)

Judge 1

	w	x	y	z
w	-	1	1	1
x	0	-	1/2	1
y	0	1/2	-	1
z	0	0	0	-

Judge 2

	w	x	y	z
w	-	0	0	0
x	0	-	1	1
y	0	0	-	1/2
z	0	0	1/2	-

Judge 3

	w	x	y	z
w	-	1	0	1/2
x	0	-	0	0
y	1	1	-	1
z	1/2	1	0	-

Judge 4

	w	x	y	z
w	-	1	1	1
x	0	-	0	0
y	0	1	-	1
z	0	1	0	-

Judge 5

	w	x	y	z
w	-	0	0	0
x	0	-	0	0
y	0	0	-	0
z	0	0	1	-

Summed Frequency Matrix

	w	x	y	z
w	-	3	2	2 1/2
x	0	-	1 1/2	2
y	1	2 1/2	-	3 1/2
z	1/2	2	1 1/2	-

When the five judges' frequency matrices were summed, the result was the summed frequency matrix which summarized all of the votes cast for each alternative.

Next, the complementary relationship of each alternative pair (x,y or y,x) was compared to determine which relationship x > y or y > x was dominant for the summed frequency matrix. In Example 9, for w and x, w,x = 3 while x,w = 0; therefore, w dominates x (wMx). The preference was constructed next giving 1's for the most preferred (dominant) complement of a pair and 0's for the least preferred (dominated) alternative of the pair. Note that z and x in the summed frequency matrix were tied with two votes each. For the tied relations, such as z and x in Example 9, 0.5 was assigned to each alternative (z,x = 0.5 and x,z = 0.5). The sum of the majority victory and tie scores for each alternative (each row) in the preference matrix provided a total preference score. If the alternatives are then ranked in descending order of the preference row sums, a, the result is the Shannon Majority Rule Method rank order, as in Example 9:

(Example 9)

Preference Matrix

	w	x	y	z	Row Sum a
w	–	1	1	1	3
x	0	–	0	1/2	1/2
y	0	1	–	1	2
z	0	1/2	0	–	1/2

The Shannon Method Rank Order is w > y > x = z

In summary, the Shannon Majority-Rule Method has been selected and demonstrated as the preferred method for aggregation of multiple criteria rank-ordered R & D projects into a single rank-ordered list of R & D projects. The Copeland and both Borda methods have had previous usage and are of analytical interest. Since they can be computed as simple deviations from the Shannon method, the Copeland and both Borda methods will be further explored in this research.

CHAPTER IV. AGGREGATION MODEL

In this chapter, a model for the aggregation of multiple ordinal rank orders is developed. The model includes the ancillary features of input conversions, weighting, fuzzy set orders, and evaluation tests.

A. Assumptions

The model for the aggregation of R & D project rank orders into a priority list was based on several restrictive assumptions as follows:

1) The number of judges and the number of alternatives both exceeded two.

2) The individual judge ranking actions were one-time decisions, not sequential steps, nor revisions.

3) There was no dialogue or coalition between the judges.

4) The set of individual judge sublists was multiple criteria in that each judge's rank order was based on his own preference criteria for the R & D projects or product requirements.

5) When a judge's ranking of several alternatives was indifferent, it was a transitive indifference. When the ranking of more than two alternatives was indifferent, this was a restrictive assumption.

6) The individual judge's sublists were transitive rank orders. Each list rank ordered an alternative only once.

7) The judge's individual preference sublists were sincerely ranked. Insincere rank orders were not developed to implement a strategy to attempt to force the final aggregated answer.

8) The individual sublists were ordinal rank orders when aggregated. If one list was originally prepared with cardinal values, it was converted to an ordinal rank order before aggregation.

B. Model Core Characteristics

The central core functions of the aggregation model were sized to aggregate up to 100 full or partial length individual rank orders with a maximum of 100 different alternatives.

The individual sublists were in a standard rank-ordered format when input to the core. The standard format was a single sequence of integer index numbers which represented the relative positions of each ordinal alternative. The Shannon Majority-Rule Method, a matrix format aggregation method described in Chapter III A, Sections 4g and III B 2, was the central analysis device in the model core.

C. Model Ancillary Characteristics

A variety of ancillary features supported the aggregation model core. These features included input manipulation, weighting, fuzzy set comparisons, and aggregation evaluation. The characteristics of each ancillary feature are discussed in the following paragraphs:

1. Input Manipulation

The judge's sublists may originally be prepared as rank orderings of R & D projects or as rank orderings of required products which could be manufactured after completion of the appropriate R & D projects.

a. Projects Sublists

The individual judge's sublists of R & D projects may be ordered in several formats and can have different characteristics. The format can be hierarchical categories or cardinal lists while the

list characteristics can be complete, partial, or have a key threshold alternative separating two portions. The objective of the input features was to convert the input projects' sublists into a single rank-ordered ordinal sublist of projects for aggregation. For the larger problems, the model with automation for the conversion should significantly reduce errors.

1) Formats - Each input alternative sublist can have any of the following formats:

a) Cardinal Utility Valued - The sublist is input by a judge as a sequence of cardinal utility values, one for each alternative. These cardinal values could be performance values, unit profits, or any other values. In all cases, for this format, the values must be those in which a higher value denotes a higher rank.

The model converts the input into an ordinal rank order of alternative-identifying numbers with the order determined by the order of the cardinal utility values.

b) Ascending Values - The format values will be those in which a lower value denotes a higher rank. The model converts the input sublists into an ordinal rank order of alternative identifying numbers with the order as the inverse of the order of values. The sublist may be input as a sequence of Julian dates or years. Each alternative would have one date. The dates represent the time when the R & D Project will be completed or when the required product will be produced and operational. The sublist could be input as a sequence of projected costs for each project or required product. The format values will be those in which a lower cost denotes a higher rank. This same input feature also applies to a list that has other cardinal

utility values in which a small value denotes a higher rank. The model converts the input sublist data into an ordinal rank order of alternative identifying numbers with the index order as the inverse of the order of cardinal values.

c) Ranked Categories of Ranked Alternatives - The sublist is input as a two-tier hierarchical list in which the categories consist of groupings of alternatives. In this format, both the categories are ranked and the alternatives within each category are ranked separately. The intent is to generate a single rank-ordered list of alternatives. The category priority list is input as are the priority lists for the sets of the alternatives in each category. The model converts the hierarchical prioritized list and multiple alternative lists into a single rank-ordered list of alternative identifying numbers. The first alternatives in the final list are the rank-ordered alternatives in the highest priority category. These are followed, lower on the list, by rank-ordered alternatives in the second priority category. This process continues until all alternatives are placed in the final list.

d) Unranked Categories of Ranked Alternatives - The sublist is input as a two-tier hierarchical list in which the categories are unranked but the alternatives within each category are ranked separately. Again, the intent is to generate a single rank-ordered list of alternatives. The priority lists for the set of alternatives in each category are input. The model converts the hierarchical list and multiple priority alternative lists into a single rank-ordered list of alternative identifying numbers.

For this format, the categories are assumed to be indifferently ranked, thereby making each category equal. The first alternatives in the final list are indifferent (equal) listings of all of the first place alternatives in each category's alternative list. These are followed, next lower in rank, by the indifferent listing of all of the second place alternatives in each category's list. This process continues until all alternatives are placed in the final list.

e) Ranked Categories of Unranked Alternatives - Again, the sublist is input as a two-tier hierarchical list in which the categories consist of grouping of alternatives. In this format, the categories are ranked but the alternatives within each category are unranked. Again, the intent is to generate a single rank-ordered list of alternatives. The priority list for the categories is input. The model converts the hierarchical prioritized list and multiple alternative lists into a single rank-ordered list of alternative identifying numbers. For this format, the alternatives within each category are assumed to be indifferently ranked, thereby making each alternative equal within a category. The first alternatives in the final list are the indifferent (equal) listings of all of the alternatives of the first place category. These are followed, lower on the list, by the indifferent listing of all of the second place category. This process continues until all alternatives are placed in the final list.

f) Unranked Categories of Unranked Alternatives - Again, the input is the two-tier sublists in which the categories consist of groupings of alternatives. In this format, neither the categories nor the alternatives are ranked. Only indifferent lists

are input. The model converts the hierarchical lists into a single rank-ordered list of alternative identifying numbers. For this format, the categories are assumed to be indifferently ranked, thereby making each category equal. Also, the alternatives within each category are assumed to be indifferently ranked, thereby making each alternative equal within a category. The alternatives in the final list are the equal listing of all alternatives in all categories, since the input gives no information for any relative differentiation.

2) Characteristics - The input sublists of alternatives can also have one of the following characteristics:

a) Complete Sublist - The sublist is input as a rank-ordered sequence of alternative identifications which ranks all alternatives of interest to the aggregation problem in question. The model core accepts this sublist in its initial form.

b) Incomplete Sublist - The sublist input ranks less than all alternatives of interest to the aggregation problem. The model may accept the incomplete sublist, as is, for aggregation, or, upon command, the model may synthetically complete the sublist. The model first determines which alternatives were not included in the input sublist. The judge is assumed to be indifferent (equal) concerning the relative relationship of alternatives not included in the input. Then, as commanded, the model adds the alternatives not included as inputs equally, at one level lower rank, to create the synthetically completed rank order. If otherwise commanded, the alternatives not included as input can be added equally, at one level higher rank, to create the beginning of the synthetically completed order. These two completion alternatives represent the situation where the input

incomplete alternative's sublist is rated either as better than all of those alternatives not included or as worse than all of those alternatives not included.

c) Threshold List - The sublist is input as an incomplete rank-ordered sequence of alternative identifications. Also, input is a specific threshold key alternative and an incomplete reference rank-ordered list. The model will merge the input sublist with the reference list using the rank policy that all of the input list is preferred to the key threshold alternative in the reference list and all alternatives below the key alternative. By command, the direction may be reversed where all of the input list is designated as inferior to the key threshold alternative in the reference list and all alternatives above the key alternatives. To avoid intransitivites, the combined list is searched for any duplicate occurrences of the same alternative. The assumption is made that if duplicates occur in a rank order, the higher ranked occurrence of an alternative will prevail. Therefore, the model eliminates the lower ranked of all duplicates found in the combined list. The final list is a sequence of nonduplicating alternative identification numbers.

b. Requirements Sublist

The individual judges may rank order lists of requirements for products which cause R & D projects, instead of being the R & D projects. There are not necessarily one-to-one relations between requirements and projects. One project may directly benefit several products or one product requirement may necessitate projects in several areas of R & D.

1) Formats and Characteristics - All of the input formats and characteristics of this Chapter, Section C, for R & D projects apply equally for requirements. The alternatives in the inputs can represent projects or requirements. The model is designed to accomplish all forms of input conversions for projects independently of requirements. But before the model can aggregate project rank orders, all requirement alternatives must be translated to projects.

2) Translation to Projects Sublists - Requirement alternatives are translated to project alternatives through a translation index in the model. The translation index file consists of equivalency statements between requirements and projects. When one requirement is equivalent to more than one project, the several joint projects are each assumed to be equally related to the requirement. Also, when several requirements are jointly equivalent to one project, the several joint requirements are each assumed to be equally related to the project.

The model steps through the requirement alternative in a converted input sublist one-by-one. Each project in the translation index replaces its equivalent requirement in the rank-ordered sublist. Where multiple equivalency occurs, several projects equally replace one requirement. A new project rank order is thus constructed. But this new project rank order probably has duplicates, triplicates, etc., of the same project alternative. Again, the assumption is used that an R & D project's priority level should be derived from the highest level at which it is required. Therefore, the model locates all replications of each alternative and eliminates all but the highest

occurrence of each alternative. This eliminates intransitive inputs into the aggregation matrix.

Due to the multiplicity in the translation index, the number of projects in a translated rank-ordered sublist is unknown until after each list has been translated and replications eliminated. When completion of incomplete input lists is required, this is accomplished in the model for requirements before the translation step. This sequence of completion, then translation, is derived from the initial model assumption concerning the independence of each judge. In the laboratory director scenario, the advisors are independent and frequently submit their rank-ordered sublists by mail. Also, the translation index equivalences probably will be developed by laboratory R & D managers after comparisons of the requirements with technology assessments. An advisor (judge) who ranks requirements has only an indirect relation to the R & D projects. Therefore, if his sublists are incomplete, they should be only synthetically completed by the remainder of the product requirements considered for the aggregation.

2. Weighting Methods

The relative importance and authenticity of each sublist and/or alternative may be included in the aggregation model by one or more weight factors applied to the rank-ordered sublists during the aggregation process. Two general types of weight application methods are in the model. The decision-maker method applies weight factors quantified by the decision maker responsible for the aggregation. The judge self-evaluation method applies weight factors quantified by each of the judges.

a. Decision-Maker Weighting Methods

The aggregation model has a variety of weighting mechanisms available to the decision maker. Weights can be applied to each alternative as w_i and/or to the rankings of each judge as w_j. The same w_i value is applied to a specific alternative every time it appears in the frequency matrices. The factors, w_i and/or w_j, are applied as multiplicative linear weighting factors, exponential weighting factors, additive weighting factors, or logarithmic weighting factors. Before applying any specific decision-maker weight, the 1/2 tie-generated frequency scores and the 1 frequency scores are normalized by multiplying all judge frequency matrices elements by a uniform 4 factor. This uniformly applied 4 inflates the vote counts, but eases computation by eliminating complications caused by exponentials of fractions reducing all values. Without the 4 bias factor, the logarithmic weight default value of 1 would drive 0 cell values into an undefined status. This normalization is beneficial for the exponential and logarithmic weights. For the Shannon Majority-Rule method, which scores pair majorities, this constant 4 permits w_i and $w_j > 1$ to be uniformly applied. The decision-maker weighting methods in the model are to follow.

The choice of a weighting method is a subjective choice of the decision maker. The multiplicative and exponential methods are expected to be used most often. Only one method is used for one aggregation.

1) Alternative Multiplicative Weights - This is a linear multiplicative method where each alternative, i, is assigned a weight value, w_i. The w_i value is multiplied by the i th alternative entries in the judges' frequency matrices in the Shannon Majority-Rule method. If x_{ik} is the cell value in the judge frequency matrix at the

intersection of the i th row and the k th column, then the weighted value for x_{ik} becomes $X_{ik}' = W_i X_{ik}$.

2) Combined Judge and Alternative Multiplicative Weights - This is a linear multiplicative method with dual weighting factors. For one factor, each alternative, i, is assigned a weight value, w_i. For the second factor, each judge's total sublist of alternatives are assigned another weight, w_j. The w_i and w_j values are both multiplied by the i th alternative entries in the judges' frequency matrices. If x_{ik} is the cell unweighted value for the i,k intersection, the weighted value, x_{ik}', becomes

$$x_{ik}' = w_j w_i x_{ik} \quad .$$

3) Alternative Exponential Weights - This is an exponential method where each alternative, i, is assigned a weight value, w_i. The w_i value is applied as the exponential power of the i th alternative entries in the judges' frequency matrices. If x_{ik} is the cell value for the i,k intersection, the weighted value, x_{ik}', becomes

$$x_{ik}' = (x_{ik})^{w_i} \quad .$$

4) Combined Judge and Alternative Exponential Weights - This is an exponential method with dual weighting factors. Each alternative, i, is assigned weight value, w_i, and each judge's total sublist alternatives are assigned another factor, w_j. The w_i and the w_j values are jointly applied as exponential powers to the i th alternatives in the judges' frequency matrices. If x_{ik} is the cell value for the i,k intersection, the weighted value, x_{ik}', becomes

$$x_{ik}' = (x_{ik})^{w_i w_j} \quad .$$

5) Combined Alternative Multiplicative and Judge Exponential Weights - This is a multiplicative and exponential method with dual weighting factors. Each alternative, i, is assigned weight factor, w_i, and each judge's total sublist alternatives are assigned another weight, w_j. The w_i and the w_j values are jointly applied with the w_i as multiplicative and the w_j as an exponential power to the i th alternative in the judges' frequency matrices. If x_{ik} is the cell value for the i,k intersection, the weighted value, x'_{ik}, becomes

$$x'_{ik} = w_i(x_{ik})^{w_j} \quad .$$

6) Combined Alternative Exponential and Judge Multiplicative Weights - This is a multiplicative and exponential method with dual weighting factors. Each alternative, i, is assigned weight factor, w_i, and each judge's total sublist alternatives are assigned w_j. The w_i and w_j values are jointly applied, with the w_j as multiplicative and the w_i as an exponential power to the i th alternative in the judges' frequency matrices. If x_{ik} is the cell value for the i,k intersection, the weighted value, x'_{ik}, becomes

$$x'_{ik} = w_j(x_{ik})^{w_i} \quad .$$

7) Combined Alternative and Judge Additive Weights - This is an additive method with dual weighting factors. Each alternative, i, is assigned weighting factor w_i, and each judge's total sublist alternatives are assigned w_j. The w_i and w_j values are jointly applied as additions to the i th alternatives in the judges' frequency matrices. If x_{ik} is the cell value for the i,k intersection, the weighted value, x'_{ik}, becomes

$$x'_{ik} = x_{ik} + w_i + w_j \quad .$$

8) **Combined Alternative and Judge Logarithmic Weights** - This is a logarithmic method with dual multiplicative weighting factors. Each alternative, i, is assigned weighting factor w_i and each judge's total sublist alternatives are assigned w_j. The w_i and w_j values are jointly applied as multiplicative to the i th alternatives in the judges' frequency matrices. Then the logarithm is taken. w_i and w_j are restricted to values equal to or greater than one. If x_{ik} is the cell value for the i,k intersection, the weighted value, x'_{ik}, becomes

$$x'_{ik} = \log [w_i w_j x_{ik}] \quad .$$

b. Judge Self-Evaluation Weighting Methods

The quality of a rank-ordered priority aggregation can be significantly influenced by the variation in each judge's knowledge of each alternative he chooses to rank. The aggregation model for this research has provisions for each judge to rate his own expertise about each alternative in regard to the criteria being used for the ranking.

The self-rating scale may be selected differently for each sublist. The scale is restricted in that the poorest rating must be zero. The best rating may be any real number, except zero. Each judge, when ranking the alternatives, also assigns a self-rating value which represents an estimate of that judge's expertise on that alternative with regard to that judge's ranking criteria. The model converts the scale used to a 0 - 1 scale. The converted self-expertise rating is multiplied by the alternative scores for the judge's frequency matrix as if the self-rating were a multiplicative weight.

The model also provides for those aggregation problems where the decision maker chooses to disregard alternative rankings by judges who have low self-evaluation ratings. The decision maker chooses a threshold

rating, on the 0 - 1 scale. All judged alternatives with converted self-ratings below the threshold value are purged from the judge sublists. The aggregation problem continues with the alternatives self-rated above the threshold. The remaining calculations are those of aggregating partial sublists with multiplicative weightings for each alternative.

3. Preference Scoring Constants

Most of the majority-rule methods described in Chapter III, Section A4 and the other literature surveyed can be implemented by similar array type computations. But the preference scoring constants of the methods differ. For example, the Shannon Majority-Rule Method uses

$$x(i,k) = \begin{cases} 1 \text{ if } i > k \\ 1/2 \text{ if } i = k \\ 0 \text{ if } k > i \end{cases}$$

for the frequency matrix and

$$x(i,k) = \begin{cases} 1 \text{ if } iMk \\ 1/2 \text{ if } i = k \\ 0 \text{ if } kMi \end{cases}$$

for the preference matrix. On the other hand, the Copeland Method uses

$$x(i,k) = \begin{cases} 1 \text{ if } i > k \\ 0 \text{ if } k > i \end{cases}$$

for the frequency matrix (ties not permitted) and

$$x(i,k) = \begin{cases} 1 \text{ if } iMk \\ 0 \text{ if } i = k \\ -1 \text{ if } kMi \end{cases}$$

for the preference matrix. Table 3 presents the matrix scoring constants for the seven majority-rule methods described in Chapter III, Section A4.

To permit correlation with literature examples and to allow studies of the effect of the scoring constants, the model is capable of aggregating the rank orders with any of the four options:

Frequency matrices with scoring constants of 1, 1/2, 0 or 1, 0, -1 and preference matrix with scoring constants of 1, 1/2, 0 or 1, 0, -1.

TABLE 3. MAJORITY-RULE METHODS MATRIX SCORING CONSTANTS

Majority Rule Method	Scoring Constants					
	Frequency Matrix			Preference Matrix		
	i > k	i = k	k > i	iMk	i = k	kMi
1. Borda	1	1/2	0	N/A	N/A	N/A
2. Adjusted Borda	1	1/2	0	N/A	N/A	N/A
3. Condorcet	1	1/2	0	1	1/2	0
4. Black's Simple Majority	1	1/2	0	1	1/2	0
5. Dodgson	1	N/A	0	1	1/2	0
6. Copeland	1	N/A	0	1	0	-1
7. Shannon	1	1/2	0	1	1/2	0

Note: N/A denotes that the constant for the stated conditions is not applicable.

4. Fuzzy Set Rank Orders

a. Background

The literature survey for this research disclosed that a new branch of social choice theory has been emerging since 1974. Blin [70, 71], in 1974, proposed that fuzzy set theory be applied to the social choice problem of determining group preference. Blin [71] said that in a fuzzy set problem, multiple observers' opinions are pooled and somehow aggregated to reach a consensus over some well-specified event. The model for this research had a limited capability to study the fuzzy set rank orders that could be obtained from the summed

frequency matrix of the Shannon Majority-Rule method. Where permitted by the problems, the aggregate fuzzy set rank order will be computed and compared with the Shannon method preference matrix rank orders.

Blin [70] explained that strict fuzzy pair preferences could be assigned a value of 1. The strict reciprocal preference would be assigned a 0. Those less strict would be assigned preference fuzzy values between 0 and 1. When n individual fuzzy matrices were summed, the resulting value in the cell is nk where k fell between 0 and 1.

The cells in the Shannon Method's summed frequency matrix had similar values which could be written as nk where n is the number of judge frequency matrices summed and k is an average score for the judges' preference with k between 0 and 1. Therefore, in this model, the Shannon Method summed frequency matrix divided by the total number of judges became the fuzzy set matrix.

b. Analytical Method

The fundamental definitions of a fuzzy rank relation R were clearly summarized by Buckles [83]. He pointed out that for complementary cells, x_{ij} and x_{ji}, an additional fuzzy set requirement was that $x_{ij} = 1 - x_{ji}$. For the fuzzy matrix complement cell values to sum to 1, the Shannon method rank-order sublists must be complete lists of unweighted alternatives. Also, Buckles presented the fuzzy set difference definition, in which the fuzzy set matrix is R and its transpose is R^T. The difference matrix definition is

$$R - R^T = \begin{cases} u(x,y) - u(y,x), \text{ if } u(x,y) > u(y,x) \\ 0 \text{ otherwise} \end{cases}$$

where for u(x,y) quantities, x is preferred to y.

To obtain fuzzy preferences, the steps are as follows [83]:

Step 1: Find the set difference, $R - R^T$.

Step 2: Determine the portion of each alternative that is not dominated.

Let $X^{ND}_{Col\ A} = 1 - \max (x_{1,\ Col\ A},\ X_{2,\ Col\ A}, \ldots X_{n,\ Col\ A})$

which means the nondominated value for an alternative, a, in the $(R - R^T)$ matrix is equal to one minus the greater of the values in Column A.

Step 3: The rank order of the fuzzy set is then the rank order of the X^{ND} values in descending order. As an example, let R, the fuzzy set matrix be

$$R = \begin{bmatrix} 0 & 0.9 & 0.2 \\ 0.1 & 0 & 0.7 \\ 0.8 & 0.3 & 0 \end{bmatrix} .$$

The transpose would be

$$R^T = \begin{bmatrix} 0 & 0.1 & 0.8 \\ 0.9 & 0 & 0.3 \\ 0.2 & 0.7 & 0 \end{bmatrix} .$$

And the difference set would be

$$R - R^T = \begin{bmatrix} 0 & 0.9 & 0.2 \\ 0.1 & 0 & 0.7 \\ 0.8 & 0.3 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0.1 & 0.8 \\ 0.9 & 0 & 0.3 \\ 0.2 & 0.7 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0.8 & 0 \\ 0 & 0 & 0.4 \\ 0.6 & 0 & 0 \end{bmatrix} .$$

Then the nondominated values become

$$X_1^{ND} = 1 - \max (0.6) = 1 - 0.6 = 0.4$$

$$X_2^{ND} = 1 - \max (0.8) = 1 - 0.8 = 0.2$$

$$X_3^{ND} = 1 - \max (0.4) = 1 - 0.4 = 0.6 \quad .$$

The rank order for the fuzzy set values is then

0.6 > 0.4 > 0.2 and the order is 3 > 1 > 2.

Buckles [83] summarized Bezdek, et al. [60] scalar measures of fuzzy set matrices, R. The two measures are the average fuzziness in R, F(R), and the average certainty in R, C(R). The average fuzziness,

F(R), was proportional to R's fuzziness or uncertainty about pairwise rankings. The average certainty in R, C(R), averaged the individual dominance of each district pair of rankings and was proportional to the overall certainty in matrix R. The equations for F(R) and C(R) are:

$$F(R) = \frac{tr(R^2)}{\frac{n(n-1)}{2}}$$

where $tr(R^2)$ is the trace of the matrix R^2

$$C(R) = \frac{tr(RR^T)}{\frac{n(n-1)}{2}}$$

where $tr(RR^T)$ is the trace of the matrix (RR^T). Also,

$$F(R) + C(R) = 1 \quad .$$

In summary, for complete aggregation problems, the model calculated the fuzzy matrix, the fuzzy set rank order, and the scalar measures F(R) and C(R). But the model did not have the sensitivity recommended by Bezdek [60]. Bezdek said that each judge should make each vote fuzzy. Instead the model used the total vote for an alternative as fuzzy. Only the judge self-evaluation model option provided single-vote fuzziness.

5. Aggregation Evaluations

The evaluation methods selected had to accommodate the several ($n \geq 3$) rank orders being aggregated, and the several aggregated rank order outputs from the different methods. Three Kendall [22] methods were chosen for the evaluations:

a) Kendall's Coefficient of Concordance Test.

b) Kendall's Circular Triads Analysis.

c) Kendall's Rank Order Consistency Analysis and Test.

The implementation of each into the model will be discussed in the next sections.

a. Coefficient of Concordance Test

1) The Coefficient - Kendall's Coefficient of Concordance, W, was chosen as a measure of the relation among several rankings ($n \geq 3$) of alternatives. Arrow [1] said that Kendall's statistic W may be used in the same situation in which Friedman's [132] two-way analysis of variance by ranks test statistic was available. Conover [9] said, further, that Kendall's W was probably intended as a measure of agreement in rankings rather than as a test statistic. This interpretation of Kendall's W coincided with the needs of this research model. The coefficient of concordance was applied in two ways. First, the method measured the agreement among the judges' sublists. Second, the method measured the agreement between combinations of aggregated rank orders obtained with different majority-rule methods, i.e., Borda, Adjusted Borda, Fuzzy Set, and Shannon. The first application provided an indication of the agreement and divergence of the judges, while the second application provided a measure of the agreement between the final rank order results, not just the winning alternative, from different majority-rule methods. The Kendall's W method was limited to sets of rank orders that had the same length. Therefore, if partial and complete rank orders were aggregated together, a Kendall's W measurement cannot be calculated unless the incomplete rank orders were synthetically completed.

The rationale of the coefficient of concordance, W, was to serve as an index of the divergence of the actual agreement shown in the data from the most perfect agreement [Seigel, 37].

To compute W, first array the rank orders in a table, with the M judge ranks listed in rows and alternatives in columns. Next find the sum of the ranks, R_j, in each alternative column. The mean sum, $\bar{R}_j$, is calculated by summing the R_j values for all alternatives and then

dividing by the number, N, of different alternatives. Next, the deviation from the mean $|R_j - \bar{R}_j|$ is calculated for each alternative. Then the square of these R_j deviations is summed into an S value. S, therefore, is stated as

$$S = \sum_{j=1}^{N} \left[R_j - \frac{\Sigma R_j}{N} \right]^2 .$$

Tied alternatives in a ranking caused complications in Kendall's W computations. Excess numbers of tied ranks in an aggregation tended to depress the value of W. A correction was available, Kendall [22], to adjust this effect of excessive tied rankings.

The tied ranking correction and the squared sum of deviations, S, are used in the coefficient of concordance, W, equation

$$W = \frac{S}{\left(\frac{1}{12}\right) M^2 \ (N^3 - N) - M \sum_{i=1}^{M} T}$$

where M is the number of judges and N is the number of alternatives. If there are no significant tied ranks, W is

$$W = \frac{S}{\left(\frac{1}{12}\right) M^2 \ (N^3 - N)}$$

2) The Test - Kendall [22] developed methods and special small N value probability tables to test the hypothesis H_o: there was perfect disagreement between the judges (there was no concordance between judges). The test for H_o varied depending on the value of N (the number of alternatives). W varied from 0 to 1. It would be 1 when the ranks assigned by each judge were exactly the same as those by other judges. W would be 0 when there was maximum disagreement among the judges. The test methods also varied with the values of M. The method steps and tables are in Dobbins [14].

b. Circular Triads Analysis

Kendall's Circular Triads Analysis [22] was chosen as a measure of the acyclity of the pair majorities in the preference matrix of the Shannon method. In preference matrices of more than three alternatives, it was possible to have the majority preferences of three alternatives aligned to be circular triads. For an example, Kendall presented a preference matrix example ([22], pp. 145) which would have a Shannon majority-rule aggregate rank order of A = C > B = E = F > D. When analyzed internally, it had five circular triads: ACDA, ABDA, AEDA, AFDA, and BEFB. Triads were counted because, for example, any circular tetrads must contain two circular triads. Kendall further proved that the maximum possible number of circular triads is

$$\frac{(N^3 - N)}{24} \text{ if N (number of alternatives) is odd,}$$

and it is

$$\frac{(N^3 - 4N)}{24} \text{ if N is even.}$$

The minimum number of triads is zero. He further proved that the maximum and minimum number of triads can be attained by arrangement of preferences. Kendall's equation for d, the number of circular triads in a preference matrix, consisted of the terms N (number of alternatives) and a_i, the sum of the rows of the preference matrix. The equation for d is

$$d = \frac{1}{6} N (N - 1) (N - 2) - \frac{1}{2} \sum_{i=1}^{N} a_i (a_i - 1) \quad .$$

The Kendall derivation of d was based on rank orders without tied pairs (indifference). When a preference matrix had tied pairs, it caused pairs of a_i terms that have fractions. The fractions were always one-half, i.e., a_i is 1.5, 3.5, 6.5, 7.5. When tied pairs existed, the

sum of the a_i was not necessarily

$$\binom{N}{2} = \binom{2}{2,\ N-2} = \frac{N!}{2!\ (N-2)!} = \frac{N\ (N-1)}{2}$$

which is the sum of a_i for integer valued, no tied pairs, preference matrices. To resolve this problem, the model for this research bracketed the possible d values if fractional pairs of a_i's exist. The steps of the d bracketing method are:

Step 1: Arrange the a_i row totals in order of their value.

Step 2: Count the number of fractional a_i row totals.

Step 3: Round the upper one-half of each pair of the fractional a_i values upward to their next larger integer values.

Step 4: Round the lower one-half of each pair of fractional a_i values downward to their next smaller integer.

Step 5: Verify that the sum of the rounded a_i's equals 1/2 N (N - 1).

Step 6: Calculate a d value for this rounded set of a_i values. Label this the "lower d" since it will give the lower value of zeta, the coefficient of consistency yet to be described.

Step 7: Return to the ordered unrounded a_i's and round the upper one-half of each pair of the fractional a_i values downward to their next smaller integer values.

Step 8: Round the lower one-half of each pair of the fractional a_i values upward to their next larger integer values.

Step 9: Verify that the sum of the second rounded a_i's equals 1/2 N (N - 1).

Step 10: Calculate a d value for this second rounded set of a_i values and label this the "upper d."

Step 11: Average the "lower d" and upper d" to form an approximate d for the matrix with the tied pairs. The approximate d values are not necessarily integers, but may be rounded to an integer.

c. Coefficient of Consistency

Kendall [22] extended the number of circular triad analyses to a coefficient of consistency, zeta, which related the calculated number of circular triads, d, to the maximum number possible: $1/24\ (N^3 - N)$ if N is odd of $1/24\ (N^3 - 4N)$ if N is even. The equation for the coefficient of consistency is

$$\text{zeta} = \begin{cases} 1 - \dfrac{24d}{N^3 - N}, & \text{if } N \text{ is odd} \\[2ex] 1 - \dfrac{24d}{N^3 - 4N}, & \text{if } N \text{ is even.} \end{cases}$$

For no inconsistencies (no circular triads), zeta is unity. As the number of circular triads increases, zeta approaches zero.

The test method for Kendall's coefficient of consistency varies with N. Special tables modified from Svestka [40] and Kendall [22], are in Dobbins [14]. The hypothesis tested is H_o: there is no consistency in the aggregated rank order. The test for H_o again varied depending on the value of N (the number of alternatives). Begin with a calculated zeta, and follow the procedures in Dobbins [14].

D. The Model

The generalized model flow to perform the rank-ordered prioritized list aggregation and analysis for this research is illustrated in Figure 2. The basic steps in the process are as follows:

1) The aggregation problem is defined, formulated, and input into the computer code in a form consistent with the model.

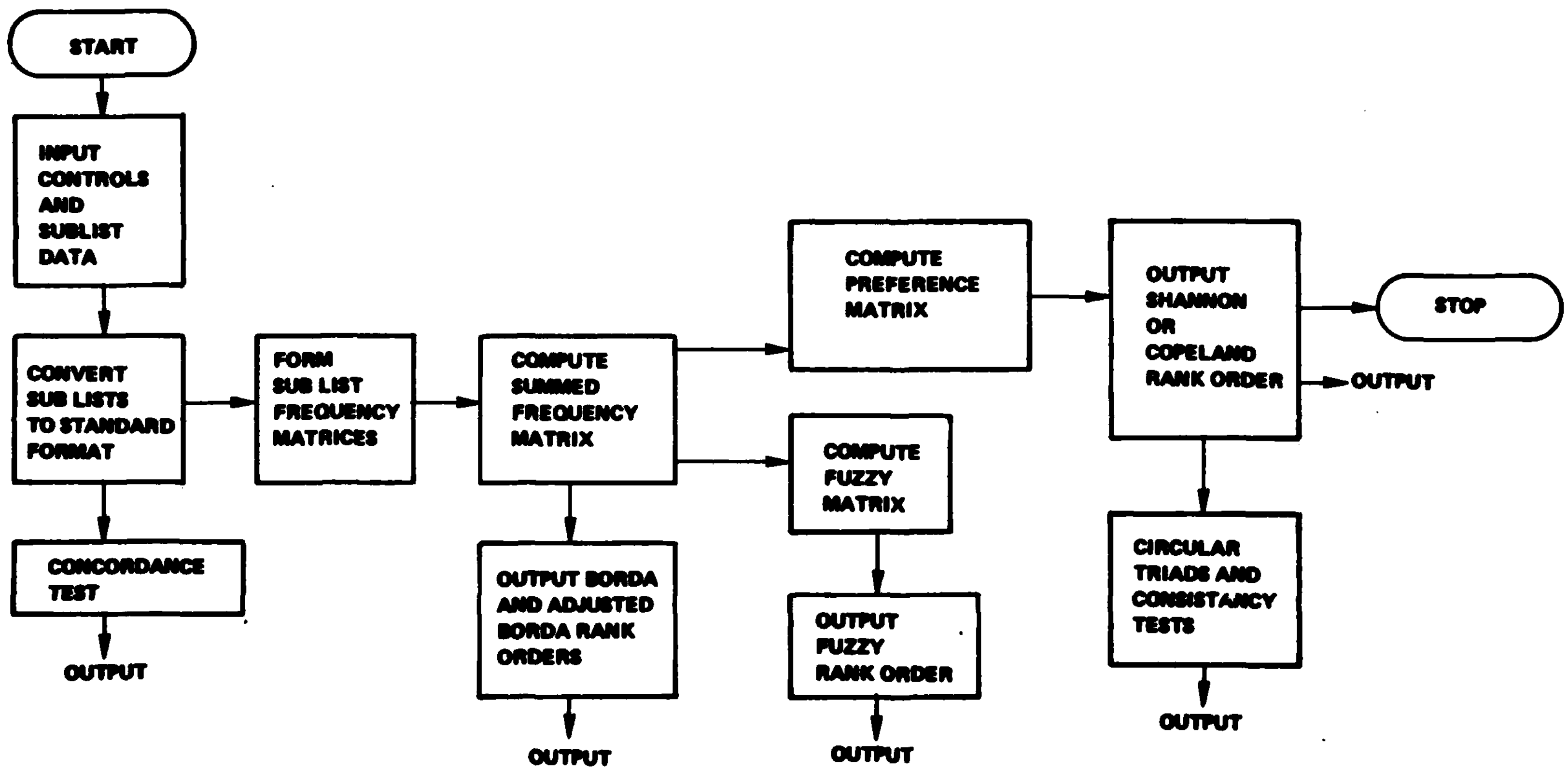


Figure 2. The generalized model.

2) The computer program converts all rank-ordered sublists into its standard format.

3) Each rank-ordered sublist is formed into sublist frequency matrices.

4) The standardized sublists are analyzed to determine and significance test their Kendall's concordance, W.

5) The sublist frequency matrices are summed.

6) The row and column totals of the summed frequency matrix are used to compute the Borda and Adjusted Borda rank orders.

7) Compute the preference matrix from the summed frequency matrix.

8) Compute the Shannon method or Copeland method output rank order from the sum of the rows of the preference matrix.

9) Compute and test the circular triads and consistency of the preference matrix.

10) Compute the fuzzy matrix and fuzzy rank order from the summed frequency matrix.

The preceding generalized model was used to develop the model explained in the following two chapters and used for the R & D Project Prioritization Study.

CHAPTER V. MODEL IMPLEMENTATION

This chapter gives a summary discussion of the computer code implementation of the ordinal rank order aggregation described in Chapter IV. The programming computer code structure is described and the output data format is briefly discussed. The computer code is listed in the appendix. Extensive instructions and example computer problems were documented in detail. However, because of the large volume of material, all but the code listing are published in another report, Dobbins [14].

A. Model Development Overview

The aggregation of multiple criteria rank-ordered priorities model presented in Chapter IV was implemented in Extended FORTRAN, Version 4, on the CDC CYBER 74. The computer used the NOS/BE executive program and has a 400,000 octal space capacity in its central memory. The computer facility is located in the Scientific and Engineering Division of the Management Information System Division of the US Army Missile Command (MICOM). Redstone Arsenal, Alabama.

The code was developed as an experimental program; therefore, achievement of its maximum matrix size was not a major consideration. The full potential for this priority rank-ordered method can be realized when its matrix dimensions are re-optimized for the applied problem of R & D project prioritization. Instead of the present 100 x 100 matrix dimension, the more practical dimension may be on the order of 50 x 200 (50 judges with up to 200 alternatives).

The code design was modularized through the use of subprograms to facilitate phased development, refinements during research, verification, and validation. The thirteen subroutine programs will be described. Since the design goal of the model was to form and manipulate up to 100 x 100 element matrices and to aggregate one-time sublists, the batch processing mode of computation was chosen as the most practical.

B. Model Code Description

1. Overall Computer Model Steps

The computer model for this research requires large computer core storage space but operates very rapidly since it does not use iterative calculations. Further, the model's code design emphasized flexibility of programming options as well as future operational flexibility to input and aggregate a wide variety of sublist priority order styles generated from many ranking criteria.

The flexibility of the model encompasses the wide variety of sublist formats that have been anticipated, such as requirements lists, expected operational dates, cardinal data, and the desire to develop methodology tools to permit exploration research in such areas as fuzzy set rank orders, preference scoring constants, and comparative aggregation methodologies. The comparative methodologies include the Borda, Adjusted Borda, and the Shannon preference majority-rule methods.

a. Flow Diagram

Figure 3 contains a simplified model flow diagram. A most comprehensive module is the input subroutine. This block of the code inputs and stores the requirements-to-projects translation equivalency statements that are expected to be used for a number of runs. The input also reads and assigns the run and sublist control

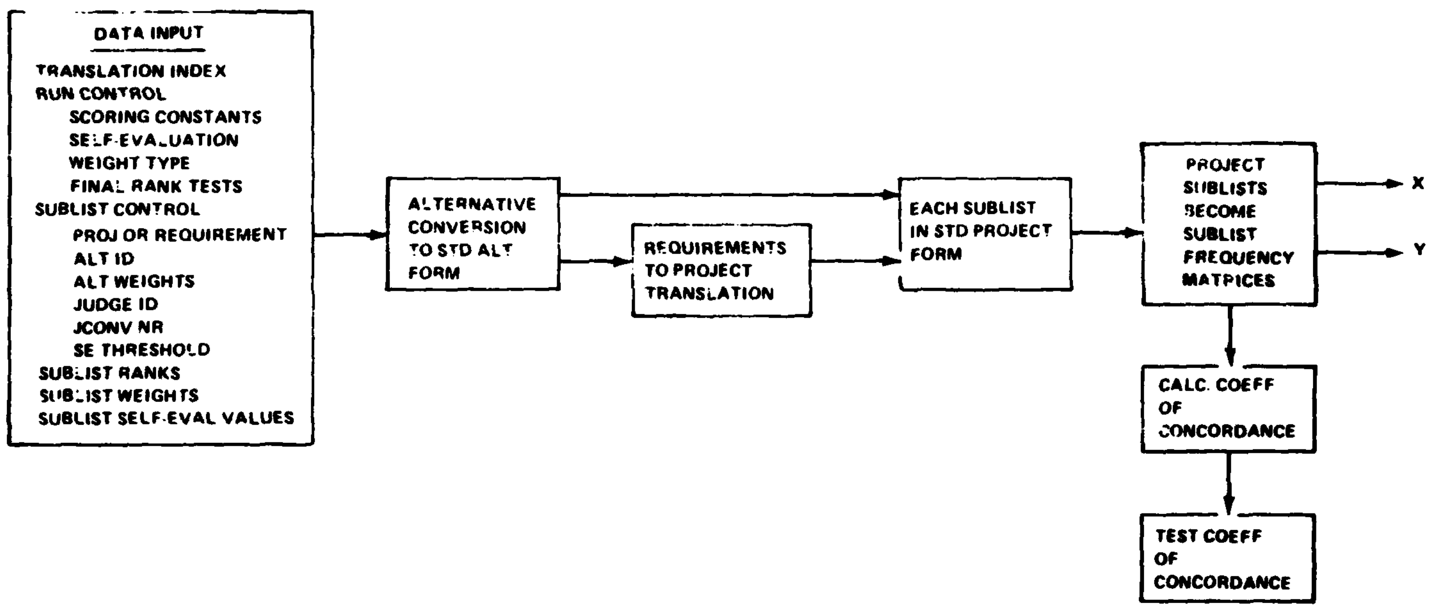


Figure 3. Simplified model flow diagram.

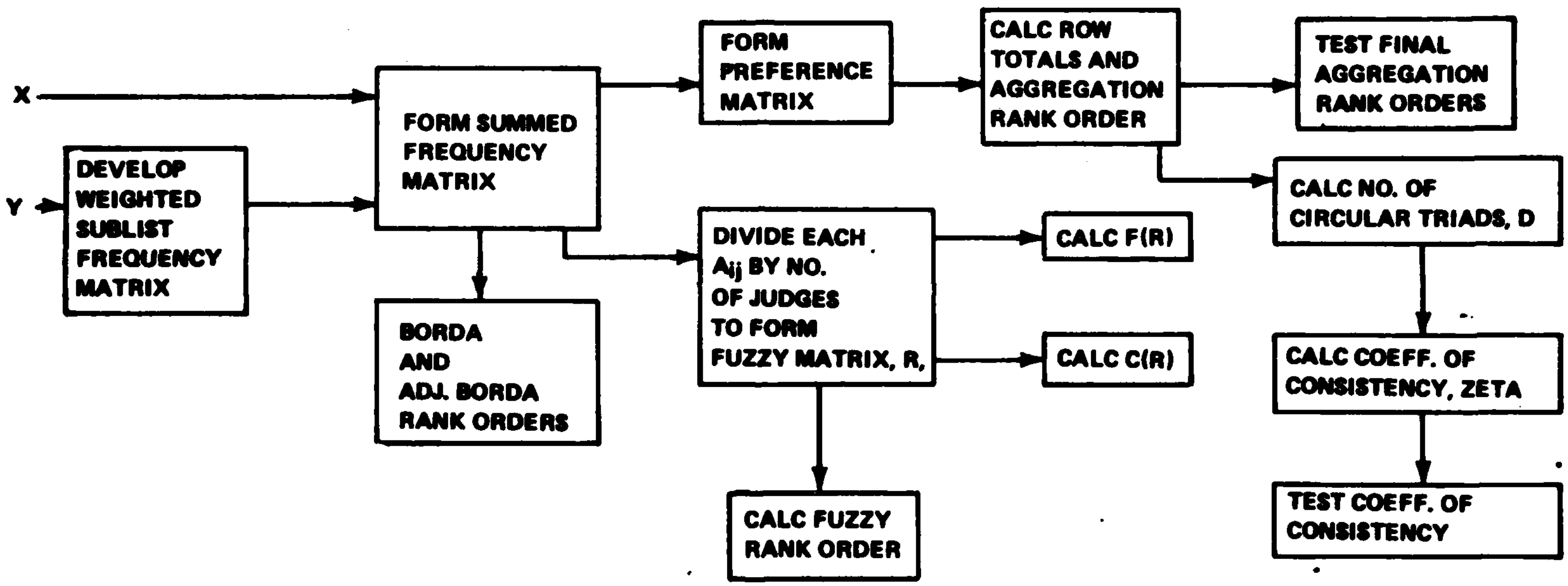


Figure 3. (Concluded).

codes for actions such as the matrix scoring constants and the weight type for the run. Typical sublist controls are alternative identifications and weights. The input also reads sublist data such as the ranks and self-evaluation values.

Found within the input subprogram, but functionally separate, is the conversion of the input sublist alternatives to a standard form. This includes conversion of cardinal score alternatives to an ordinal ranking or the conversion of categorized alternatives into single rank-ordered lists. "Alternative" is emphasized at this phase in the model since the ranked elements can be either R & D projects or product requirements.

Where alternatives are product requirements, the next phase is to translate those sublists of rank-ordered requirements into transitive sublists of rank-ordered R & D projects.

At the completion of the input, conversion, and translation phases, all sublists are ready to enter the matrix aggregation in a standard form of transitive rank-ordered lists of R & D projects.

Initially in this second phase, the sublists become sublist frequency matrices scored by the chosen constants. The same standardized sublists are used to calculate and test the statistical significance of the coefficient of concordance.

If weighting and/or judge self-evaluation are included for the run, the sublist matrix elements are next normalized by multiplication by 4 and then are weighted. At this phase, the judge self-evaluation factors become another multiplicative weight.

The sublist matrix elements, either all weighted or unweighted, are summed into the summed frequency matrix which contains the sum

of votes, or weighted votes, that each alternative received when paired against each other alternative. If there is no judge indifference, the sum of the values in each row element become the Borda count for the row (project). With or without judge indifference, the sum of the row element values minus the sum of the column element values is the Adjusted Borda count. The model rank orders these counts into the Borda and Adjusted Borda rank orders.

The summed frequency matrix element values, divided by the number of judges, becomes the fuzzy matrix, R. From R, the model calculates the fuzzy measures, F(R), and C(R), and the fuzzy rank order.

The comparison of the complement paired element values in the summed frequency matrix is the basis for the element values in the preference matrix. The preference matrix assigns scores to projects for the number of majority comparisons they win, tie, or lose. The sum of the row element values provides the aggregation count for each project. The model rank orders this count into the aggregation rank order.

The aggregation row counts also provide the inputs for the calculation of the number of circular triads, D, and the coefficients of consistency, zeta. The model tests the statistical significance of zeta.

Last, the model can compare any chosen combinations of the final rank orders (Borda, Adjusted Borda, Fuzzy, or Preference), then determine and test the Kendall's Coefficient of Concordance for these rank orders.

The appendix contains a more comprehensive model functional flow diagram which contains major decision logic nodes.

b. Subroutine Programs

The appendix contains the listing and definitions of key terms for the aggregation of multiple criteria rank-ordered

priorities computer code, developed for the report research. The code structure diagrammed in Figure 4 consists of the main program and thirteen subroutine program modules as follows:

1) Main Program - DOBBINS - The main program coordinates all mainstream processing of rank orders through the model. It calls subroutines in the proper sequence for calculations in a given run based on user and model-provided controls and data. It writes only the summed frequency matrix and the Borda-type counts and rank orders.

2) Subroutine INPUT - The subroutine reads and coordinates the input controls and data. The subroutine also converts the sublist alternative data to the standard ordinal rank-ordered format. INPUT reads the run controls and the sublist controls. It coordinates the calling of subroutine PRAM which reads sublist ranks, weights, and self-evaluation data. The self-evaluation rating full scales are converted to 0 to 1, and the specified sublist conversions JCONV 2 to 12 are performed in INPUT. This subroutine applies the self-evaluation threshold and checks all subroutines for completion. Finally it stores the converted, unweighted standard form sublists of ranked alternatives.

3) Subroutine PRAM - This is a library subroutine to enter floating point data in free format form where precise formats are not practical. In this computer model, PRAM is used to enter the sublist rank data and the self-evaluation data.

4) Subroutine REQUIR - This subroutine receives converted requirements sublists from INPUT. It compares each sublist to the translation index that has been sorted and arranged by requirement name. REQUIR then extracts the projects that match the requirement in the sublist. A project's rank order is built by insertion of the group of projects for each requirement. REQUIR purges duplications

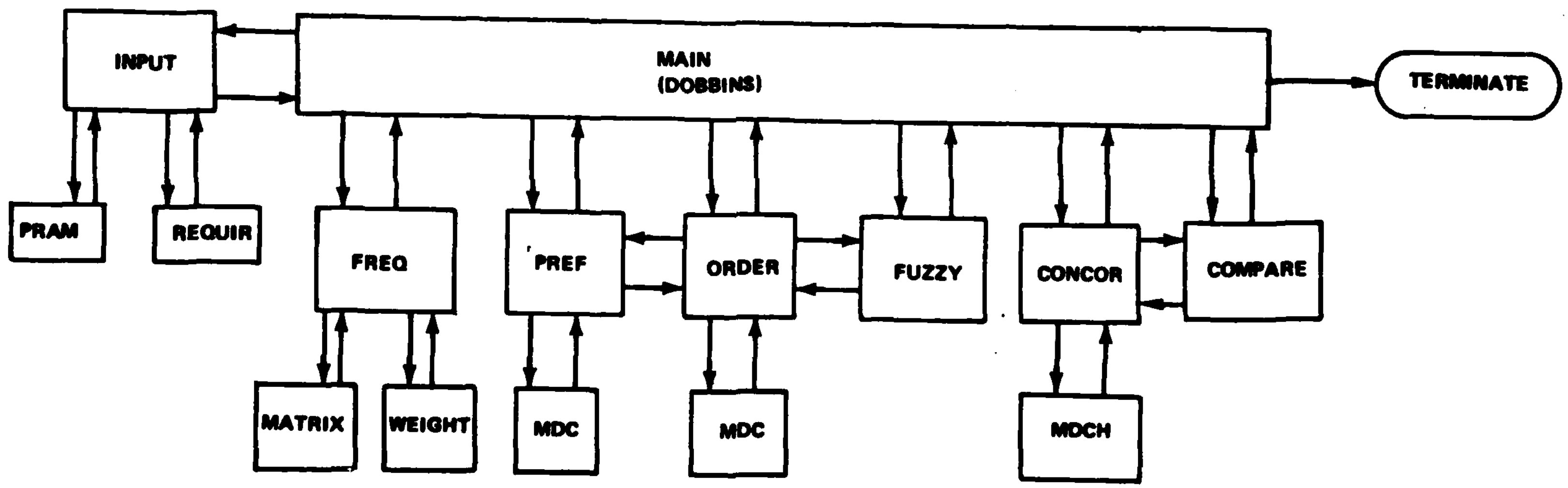


Figure 4. Subroutine module network.

from the raw projects rank order. The transitive project sublist is returned to INPUT.

5) Subroutine FREQ - This subroutine coordinates the placement of the sublists into sublist frequency matrices and the weighting of the frequency matrix elements. It further applies the self-evaluation values to the frequency matrix elements and writes the sublist self-evaluation frequency matrix.

6) Subroutine MATRIX - This subroutine forms and writes the sublist frequency matrix for each sublist. The matrices are formed using the specified matrix scoring constants.

7) Subroutine WEIGHT - This subroutine applies the specified weighting to each sublist frequency matrix element and writes the weighted sublist frequency matrix. Before any weights are applied, WEIGHT multiplies all sublist matrix elements by four.

8) Subroutine PREF - This subroutine forms and writes the preference matrix, calculates the number of circular triads, D, and the coefficient of consistency, zeta, and statistically tests zeta. The subroutine also calculates the bracket and average values for D and zeta when fractional sums occur in the preference matrix rows. All D, zeta, and test results are output by this subroutine. The matrix is formed using the specified matrix scoring constants.

9) Subroutine MDCH - This subroutine is an International Mathematical and Statistical Libraries (IMSL) program which is used for the chi-squared probability statistical tests of the Kendall coefficient of concordance and the coefficient of consistency. MDCH automatically changes to use the normal distribution approximation, Z, for the high degree of freedom with the chi-square statistic ($dF \geq 30$).

10) Subroutine ORDER - This subroutine converts a list of values for a set of alternatives into an ordinal rank-ordered list of the alternative identifications. ORDER writes the final rank-ordered list.

11) Subroutine FUZZY - This subroutine calculates the fuzzy matrix, the measures F(R) and C(R), and writes the Fuzzy project scores. This subroutine coordinates the forming and writing of the Fuzzy rank-ordered list.

12) Subroutine CONCOR - This subroutine calculates, statistically tests, and outputs the results of the coefficient of concordance for the standard sublists and the final results. CONCOR also calculates and outputs the intermediate concordance variables such as the mean and the sum of the squares of the deviations from the mean.

13) Subroutine COMPARE - This subroutine takes the final aggregate rank orders from the BORDA, ADJ. BORDA, PREFERENCE, and, if available, FUZZY methods and compares them two at a time. The pairs of aggregated lists are sent to CONCOR and evaluated, then control returns to COMPARE. When the last pair is evaluated, COMPARE returns to the main, DOBBINS, and the model terminates.

2. Input to the Computer Model

a. Overview

The inputs to this computer code have been kept relatively simple compared to the complexity of the model. Inputs are the run controls, the alternative names, the ranking and rating data which are entered in free format.

b. Option Control

Besides the alternative names and numbers, the first set of cards (one card per alternative) contain the alternative weight factors and category for that alternative. The second type of single control card has integer numerical digits to designate the type weighting technique (1 through 8), whether the sublists should be completed, the type of matrix scoring constants for both the frequency and preference matrices, the self-evaluation control for weighting or weighting and threshold elimination, and the self-evaluation scale limit value. The third type control card (one card per judge sublist) identifies the judge, designates the alternative conversion type, and gives the judge weight factors. The appendix specifies input data format in greater detail.

c. Data

The sublist ranks and the self-evaluation ratings are input as separate sets of free format cards. For the sublist rank set, the sequence of alternative numbers indicates the preference order with minus signs used to indicate equality or indifference. Each sublist ends with an asterisk. The sublist self-evaluation ratings are listed on their cards in an order corresponding to the lexigraphic order of the alternatives' identification numbers (1, 2, 3, ...etc.).

3. Output from the Computer Model

a. Aggregation Rank Orders

The primary output of the computer model for this research is the aggregated rank-ordered list of R & D projects. The Shannon majority method produces the baseline aggregation rank-ordered list for the model. For comparative purposes, the model also produces the Borda-type rank orders and the Fuzzy Matrix rank order. To permit run-by-run verification and analysis of each aggregation rank order, the model

outputs the inputs, the sublist matrices (basic, and if appropriate, the weighted and/or self-evaluation matrices) and the sum of the rows for the summed, Fuzzy, and preference matrices.

b. Evaluation Results

The model further provides the results of the evaluation of the input sublists and the aggregation rank orders. It computes, statistically tests, and prints the major steps of the Kendall coefficient of concordance evaluation of the standardized input sublists. The statistical tests conclude with statements as to whether the input rank orders are consistent at the 0.05 and 0.01 significance levels. Again, for verification and analysis of each evaluation, the model provides the rank array, the alternative sums, means, the sum of the squared deviations, tied ranking factor, and the coefficient of concordance.

The model also performs up to six Kendall's coefficient of concordance analyses of all two-rank-order combinations of the four aggregation rank orders, i.e., Shannon versus Fuzzy, Shannon versus Borda, Adjusted Borda versus Fuzzy, etc. The output details are the same as those for the coefficient of concordance evaluation for the input sublists. These evaluation data provide a measure of the agreement between the various final aggregations.

The other evaluation parameters are Kendall's number of circular triads, D, and the coefficient of consistency, zeta, which evaluate the cyclicity characteristics of the Shannon aggregation rank order. The statistical tests determine if the tested rank order could have occurred by chance, instead of by a somewhat consistent preference method.

The appendix contains the program list for the computer model.

C. Matrix and Rank Order Formation

Once the sublist rank orders are in standardized project alternative formats, the computer model forms each into a sublist frequency matrix that indicates which project is preferred over each other project by pair comparisons. The summed frequency matrix is the matrix element addition of the sublist frequency matrices. The preference matrix is formed from the paired comparisons of each of the summed frequency matrix element values.

The project scores are computed from the row and column, if appropriate, sums of the elements of the summed frequency matrix and the preference matrix. The model then places the project with the highest score highest in the rank order and repeats the search for each equal or lower scored project.

D. Ancillary Processes

1. Weighting

a. Decision-Maker Methods

The model input weighting functions are input as a control code and weighting factor data codes. The weighting factors are the weights applied to each alternative, WHI, and the weights applied to each judge, WHJ. If Alternative a has a WHI value other than 1, every time Alternative a appears in a sublist frequency matrix, it will be weighted by the factor WHI. If Judge 2 has a WHJ value other than 1, every alternative in Judge 2's sublist will be weighted by the factor WHJ. If Alternative a is in Judge 2's sublist, it will be dual-weighted by WHI and WHJ.

b. Judge Self-Evaluation Methods

The judge self-evaluation (JSE) methodology is implemented as a weighting scheme. JSE is controlled by the MATR code in

the control card, as follows:

MATR = 0 - No JSE.

MATR = 1 - The JSE factors are applied to all ranked alternatives. No threshold is applied.

MATR = 2 - The JSE factors are applied to all ranked alternatives. A threshold is applied that purges all ranked alternatives with JSE ratings below the threshold value which is input as a THLD value.

2. Evaluation Techniques

No controls are necessary to obtain evaluation of the input rank orders. The final aggregation rank orders are comparatively evaluated by the coefficient of concordance method for each combination of final rank orders that is calculated by the model. The evaluation and test calculation techniques are described in Chapter IV and Dobbins [14].

CHAPTER VI. MODEL VERIFICATION AND VALIDATION

This chapter contains the verification and validation of the computer model, including sample numerical validation problems.

A. Verification

1. Model Design and Test

Verification that the rank-ordered aggregation computer model was implemented properly in the computer code was accomplished through the modular design of the code, unit testing of each subroutine, phased buildup of the computer model with tests after each phase was added, running a series of test problems for comparisons of computer model output with hand-calculated results, and a final exercise of all options in the program.

Extensive model validation in the sense of running large aggregation rank-ordered priorities problems was not possible due to the lack of available problems with known solutions using any of the four majority-rank methods that are built into the computer model. Validation of portions of the model options against moderate-sized known problems with solutions from the literature was accomplished. Some of the special features of this model, such as weighting, fuzzy rank orders, and judge self-evaluation were validated by calculated extensions from matrix aggregation methods confirmed against the literature.

2. Verification Demonstration

The computer model's flexibility was verified and demonstrated in Dobbins (14) through the exercise of most of the computation options

for a single set of partial sublist rank orders, a set of alternative and judge weights, and a set of self-evaluation ratings.

B. Model Validation

Computer model validation was accomplished by comparing results of the computer model to results for examples found in the literature. The literature often gave only winners for the method employed. The validation was divided into areas of method emphasis in the literature examples as follows: Borda and Adjusted Borda; Borda, Condorcet, and Black; Borda, Condorcet, Black and Copeland; Copeland; Shannon preference and others; and special purpose examples to validate other model areas such as tied data and evaluation tests. Each case in Tables 4 through 9 presents selected literature examples, published results, comparable results from the computer model, and additional model results. A total of 46 literature cases was validated, but only 13 are in this report. All 46 validation cases are published in Dobbins (14).

All six tables have the same format. The left one half of each table page is quoted from the literature. First, the reference identification is listed; then the example sublist rank orders are shown. Last, key answers from the literature are given. The right one half of each table page contains results from aggregating the literature example sublists in the computer model. The upper left portion of the computer model side of the page contains the various final rank orders as computed. The upper right portion of the model side of the page contains the results of the coefficient of consistency testing of the preference matrix. The lower portion of the model side of the page contains coefficient of concordance results for the sublists and for selected pairs of final aggregation results.

For Case 1 of Table 4, Richelson presented X as the Borda winner which was in agreement with both computed Borda orders. In Case 2, the literature example gave the Borda and the Adjusted Borda counts. For Case 2, where there were no ties in the sublist ranks, the computer data fully agreed with the literature examples.

For Case 2, the asterisk (*) at the sublist concordance results denotes that the examples indicated contained repetitions of sublists. For these cases, the repetitions of sublists were input as multiplicative judge weights (Wj). The rank orders were the same but the sublist concordance data were based on single occurrences of each type of sublist. It was concluded that the model adequately represented the Borda and Adjusted Borda majority-rank methods.

For the cases of Table 5, for Borda, Condorcet, and Black method examples, the Borda results were computed, and the Condorcet results were observed by scanning the rows of the preference matrix for zeros. If a zero (other than on the main diagonal) existed, then the alternative did not have a majority over all other alternatives, which is the Condorcet criterion. The Black answer is the Condorcet winner if one exists. If a Condorcet winner did not exist, the Black winner is the Borda or Adjusted Borda winner. A strong Condorcet winner is one that beats, not ties, all other alternatives. For Table 5, computed results for both cases agreed with the literature examples. For Case 2, the Borda count values were also given in the literature and were in agreement with the computer model results.

TABLE 4. MODEL VALIDATION WITH LITERATURE - I

Borda and Adjusted Borda

Literature	
Case 1 Ref: Richelson [202] p. 42	
Qty	Sublist Orders
1	x > y > z
2	x > z > y
4	y > x > z
Answer:	Borda: X

Model Results		
Aggregation Orders		Consistency
Borda:	x > y > z	D = 0
Adj. Borda:	x > y > z	zeta = 1
Pref:	y > x > z	5% Cons: Yes
Fuzzy:	y > x > z	1% Cons: Yes

Concordance	Sublists	Borda/Pref
Mean:	14	4
Sum sq:	38	6
Coeff W:	0.388	0.75
5% Conc:	No	No
1% Conc:	No	No

TABLE 4. (CONCLUDED)

Borda and Adjusted Borda

Case 2 Ref: Black [4] p. 61

Qty	Sublist Orders
1	$A_3 > A_2 > A_1 > A_4 > A_5$
2	$A_1 > A_2 > A_3 > A_4 > A_5$
8	$A_4 > A_5 > A_3 > A_2 > A_1$
9	$A_5 > A_4 > A_3 > A_2 > A_1$
15	$A_2 > A_1 > A_3 > A_4 > A_5$

Answer: Borda Count:
$A_1 = 55$, $A_2 = 86$, $A_3 = 72$,
$A_4 = 77$, $A_5 = 60$

Aggregation Orders	Consistency
Borda Count: $A_1 = 55$, $A_2 = 86$, $A_3 = 72$, $A_4 = 77$, $A_5 = 60$	D = 0
Borda = Adj Borda: $A_2 > A_4 > A_3 > A_5 > A_1$	zeta = 1
Pref: $A_3 > A_2 > A_1 > A_4 > A_5$	5% Cons: Yes 1% Cons: Yes

Concordance

	*Sublists	Borda/Pref
Mean:	15	6
Sum Sq:	18	26
Coeff W:	0.072	0.65
5% Conc:	No	No
1% Conc:	No	No

*See the text.

TABLE 5. MODEL VALIDATION WITH LITERATURE - II

Borda, Condorcet, and Black

Case 1 Ref: Fishburn [119], p.540

Qty	Sublist Orders
1	x > y > A > B > C
1	y > A > C > B > x
1	C > x > y > A > B
1	x > y > B > C > A
1	y > B > A > x > C

Answers: Borda: y

Cordorcet: x

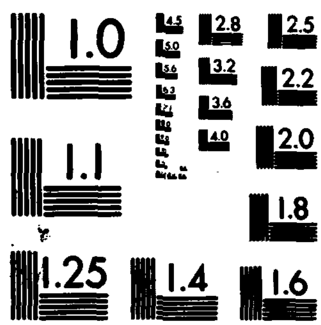
Aggregation Orders		Consistency
Borda:	y > x > A > B = C	D = 0
Condorcet:	x	zeta = 1
Pref:	x > y > A > B > C	5% Cons: Yes
Fuzzy:	x > y > C > A > B	1% Cons: Yes

Concordance

	Sublists
Mean:	15
Sum Sq:	62
Coeff W:	0.248
5% Conc:	No
1% Conc:	No

AD-A091 684 ARMY MISSILE COMMAND REDSTONE ARSENAL AL TECHNOLOGY--ETC F/G 5/1
A METHODOLOGY FOR AGGREGATION OF MULTIPLE CRITERIA RANK-ORDERED--ETC(U)
MAY 80 E B DOBBINS
UNCLASSIFIED DRSMI/RN-80-3-MR SBIE-AD-E950 034 NL

3



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

TABLE 5. (CONCLUDED)

Borda, Condorcet, and Black

Case 2 Ref: Richelson [203] p. 173

Qty	Sublist Orders
1	x > y > z > A > B > C > D
1	y > z > x > A > C > D > B
1	A > D > B > C > Z > x > y

Answers: Borda = Black: A

Condorcet: ϕ

Borda Scores: x = y = z = 11, A = 12, B = C = D = 6

Aggregation Orders		Consistency
Borda = Black:	A > x = y = z > B = C = D	D = 2
Condorcet:	ϕ	
Pref:	x = y = z > A > B = C = D	zeta = 0.857
Fuzzy:	x = y = z > A > B = C = D	5% Cons: Yes
Borda Scores:	x = y = z = 11, A = 12, B = C = D = 6	1% Cons: Yes

Concordance

	Sublists	Borda/Pref
Mean:	12	8
Sum Sq:	48	84
Coeff W:	0.190	0.875
5% Conc:	No	No
1% Conc:	No	NO

For the cases of Table 6, the Copeland results were obtained as the preference order of the model when 1, 1/2, 0 scoring constants were used for the frequency matrices and 1, 0, -1 scoring constants were used for the preference matrix. The other results were obtained as in Table 5. Cases 1 and 2 of Table 6 have total correspondence between literature and computer results from the rank-ordered aggregations. The model is validated for the Copeland method. Both cases in Table 7 have Copeland results from the literature and the model that fully agree.

Case 1 of Table 8 has full agreement between the literature examples and the model results. In Case 2, the rank orders agreed but the sublist concordance figures differed because of an error in the sum of the ranks for the F alternative. Dr. Shannon, author of Case 2, told of the error during a class lecture. Both cases in Table 8 validated the Shannon preference method in the model.

Table 9 presents special cases to validate specific functions in the model. Case 1 is a Kendall example to illustrate the ties correction calculation of the coefficient of concordance when significant ties (indifferences) were in the sublists. The literature example and the model computation of Case 1 agreed completely.

Case 2 is a Kendall example to illustrate the number of circular triads (D) and coefficient of consistency (zeta) computations. The literature example began with the preference matrix and continued to the completion of the consistency evaluation. The literature example and the model computation of Case 2 agreed.

TABLE 6. MODEL VALIDATION WITH LITERATURE - III

Borda, Condorcet, Black, and Copeland

Case 1 Ref: Fishburn [119] p. 54

Qty	Sublist Orders
1	y > z > W > T > x
1	T > W > z > y > x
2	y > x > z > W > T
2	x > T > W > z > y

Answers: Borda = Adj Borda: x

Condorcet: y

Copeland: x

Aggregation Orders		Consistency
Borda = Adj Borda:	X > T = y > z > W	D = 3.5
Condorcet:	y	zeta = 0.3
Pref:	x > y > T > z > W	5% Cons: No
Fuzzy:	y > x = T = z > W	1% Cons: No
Copeland:	x > y > T > z > W	

Concordance

	Sublists	Borda/Pref	Borda/Fuzzy
Mean:	18	6.1	6.0
Sum Sq:	16	38.5	27.5
Coeff W:	0.044	0.983	0.786
5% Conc:	No	Yes	No
1% Conc:	No	No	No

TABLE 6. (CONCLUDED)

Borda, Condorcet, Black, and Copeland

Case 2 Ref: Richelson [200] p. 336

Qty	Sublist Orders
1	x > y > A > B > C
1	y > A > C > B > x
1	x > y > B > C > A
1	C > x > y > A > B
1	y > B > A > x > c

Answers: Black = Copeland: x

Copeland Scores:

A = 0, B = -2, C = -4, X = 4, y = 2

Aggregation Orders	Consistency
Borda: y > x > A > B = C	D = 0
Black = Condorcet: x	zeta = 1
Pref: x > y > A > B > C	5% Cons: Yes
Fuzzy: x > y > C > A = B	1% Cons: Yes
Copeland: x > y > A > B > C	
Copeland Scores: A = 0, B = -2, c = -4, x = 4, y = 2	

Concordance

	Sublists	Borda/ Adj Borda	Borda/Pref
Mean:	15	6	6
Sum Sq:	62	38	36.5
Coeff W:	0.248	1.0	0.936
5% Conc:	No	Yes	No
1% Conc:	No	No	No

TABLE 7. MODEL VALIDATION WITH LITERATURE - IV

Copeland

Case 1 Ref: Richelson [200] p. 335

Qty	Sublist Orders
1	z > y > x > A
1	A > y > x > z
2	x > y > z > A

Answer: Copeland: x = y

Aggregation Orders		Consistency
Borda:	x = y > z > A	D = 0
Pref:	x = y > z > A	zeta = 1
Fuzzy:	x = y > z = A	5% Cons: Yes
Copeland:	x = y > z > A	1% Cons: Yes

Concordance

	Sublists
Mean:	10
Sum Sq:	18
Coeff W:	0.225
5% Conc:	No
1% Conc:	No

TABLE 7. (CONCLUDED)

Copeland

Case 2 Ref: Richelson [200] p. 335		Aggregation Orders	Consistency
Qty	Sublist Orders	Borda = Adj Borda: A = x = y > B	D = 1
1	x > y > A > B	Pref: A = x = y > B	zeta = 0.5
1	A > x > y > B	Fuzzy: A = x = y > B	5% Cons: No
1	y > A > x > B	Copeland: A = x = y > B	1% Cons: No

Answers: Copeland: A = x = y > B

Concordance

	Sublists
Mean:	7.5
Sum Sq:	27
Coeff W:	0.60
5% Conc:	No
1% Conc:	No

TABLE 8. MODEL VALIDATION WITH LITERATURE - V

Shannon Preference and Others

Case 1 Ref: Shannon [217] p. xviii

Qty	Sublist Orders
1	A > C > B > E > D
1	B > A > C > E > D
1	A > B > C > D > E

Answers: Borda: A > B > C > E > D

Pref: A > B > C > E > D

Coeff W = 0.84

5% Conc: Yes

Aggregation Orders	Consistency
Borda: A > B > C > E > D	D = 0
Pref: A > B > C > E > D	zeta = 1.0
Fuzzy: A > B > C = D = E	5% Cons: Yes
	1% Cons: Yes

Concordance

	Sublists	Borda/Pref	Pref/Fuzzy
Mean:	9	6	6
Sum Sq:	76	40	34
Coeff W:	0.844	1.0	0.944
5% Conc:	Yes	Yes	No
1% Conc:	Yes	Yes	No

TABLE 8. (CONCLUDED)

Shannon Preference and Others

Case 2 Ref: Shannon [217] p. xviii

Qty	Sublist Orders
1	A > C = E > B > D > G > F > I > H > J
1	B > C > A > D = E > H > F = J > G > I
1	B > A > C > D = E = F > H > I > G = J

Answers:

Adj Borda:	A = B > C > E > D > F > H > G > I > J
Pref:	B > A > C > E > D > F > H > G > I > J

Coeff W = 0.90

χ^2 cal = 24.30

5% Conc: Yes

	Aggregation Orders	Consistency
Adj Borda:	A = B > C > E > D > F > H G > I > J	D = 0.50
Prefix:	B > A > C > E > D > F > H G > I > J	zeta = 0.978
Fuzzy:	B > A = C > E = D = F = H = G = I = J	5% Cons: Yes 1% Cons: Yes

Concordance

	Sublists	Borda/Pref	Pref/Fuzzy
Mean:	16.5	11	11
Sum Sq:	629.5	328.5	244.5
Coeff W:	0.862	0.036	0.064
χ^2 =	23.3	17.9	16.1
5% Conc:	Yes	Yes	No
1% Conc:	Yes	No	No

TABLE 9. MODEL VALIDATION WITH LITERATURE - VI

Special

Case 1 (Ties) Ref: Kendall [22] p. 97

Qty	Sublist Orders
1	1 > 3 > 5 > 2 = 4 > 7 > 6 = 9 > 8 > 10
1	2 > 1 = 3 > 4 = 5 > 8 = 10 > 6 > 7 > 9
1	2 > 1 > 3 = 4 = 5 = 6 > 7 = 8 = 9 > 10

Answer: ΣT: 9.5
Sum Sq: 591
Coeff W: 0.828

Aggregation Orders	Consistency
Adj Borda: 1 > 2 > 3 > 5 > 4 > 6 > 7 > 8 > 9 > 10	D = 0.5
Pref: 2 > 1 > 3 > 5 > 4 > 6 > 7 > 8 > 9 > 10	zeta = 0.9875
Fuzzy: 2 > 1 > 3 > 4 = 5 = 6 = 7 = 8 = 9 = 10	5% Cons: Yes
	1% Cons: Yes

Concordance

	Sublists	Borda/Pref
Mean:	16.5	11
ΣT:	9.5	0
Sum Sq:	591	328
Coeff W:	0.828	0.994
5% Conc:	Yes	Yes
1% Conc:	Yes	No

TABLE 9. (CONTINUED)

Special

Case 2 Ref: Kendall [22] p. 145

Preference Matrix

Sum		A	B	C	D	E	F
4	A	-	1	1	0	1	1
2	B	0	-	0	1	1	0
4	C	0	1	-	1	1	1
1	D	1	0	0	-	0	0
2	E	0	0	0	1	-	1
2	F	0	1	0	1	0	-

Answers: D = 5, zeta = 0.375

Aggregation Orders	Consistency
Pref: A = C > B = E = F > D (Same Pref Matrix)	D = 5.0 zeta = 0.375 5% Cons: No 1% Cons: No

Concordance (N/A)

TABLE 9. (CONCLUDED)

Special			
Case 3 Ref: Kendall [22] p. 94		Aggregation Orders	Consistency
Qty	Sublist Orders	Borda = Adj Borda: C > B > E > A > F > D	D = 2
1	C > F > E > B > A > D		zeta = 0.75
1	C > A > B > F > D > E	Pref: C > B = E > A > F > D	5% Cons: No
1	B > E > D > A > F > C		1% Cons: No
1	E > C > B > A > D > F		
Answers:	Mean: 14		
	Sum Sq: 64		
	Coeff W: 0.229		

Concordance		
	Sublists	Borda/Pref
Mean:	14	7
Sum Sq:	64	68.5
Coeff W:	0.229	0.993
5% Conc:	No	Yes
1% Conc:	No	Yes

Case 3 is another Kendall example to illustrate the calculation of the Kendall's coefficient of concordance, W. The literature example and the model calculation of the mean, the square of the deviation, and the coefficient of concordance all agreed.

In summary, the literature cases and model results in the six tables represent a reasonable validation of the model.

CHAPTER VII. R & D PROJECTS PRIORITIZATION STUDY

This chapter contains a case study of the application of the ordinal rank order aggregation method and model. The problem is that of establishing the prioritization of R & D projects.

A. Overview

The objective of this research has been to develop a methodology to aggregate multiple sourced rank-ordered lists of product requirements and R & D projects into a single list of rank-ordered R & D projects. In this chapter, the methodology toward this objective will be demonstrated. The methodology will employ the aggregation computer model described in previous chapters. The Shannon majority-rule preference method will be used for the final rank-ordered list of projects. The input sublists of ordinal product requirements and R & D projects will be used and documented in this chapter. The demonstration study will aggregate at least thirteen sublists for 95 R & D projects (alternatives). The requirements sublists will rank 44 product requirements. The sublist frequency matrix and weighted sublist matrix each require seven computer printed pages for the 95 x 95 matrices. Therefore, the approximately 182-page total computation printing for the sublist matrices will not be included in this report. Complete problem computation examples can be found in Dobbins (14) for a smaller dimensioned problem. Sample sublist data and summary results will be included for the R & D project demonstration study.

To avoid any national security concerns, the names and the correct identification numbers of the individual product requirements and R & D projects will not be used. A typical, fictitious product requirement would be R601, to obtain a new hand-held, 10-km range antipersonnel, all weather rocket-propelled weapon system. A typical, fictitious R & D project would be: PGD-65, to develop the technology to locate, identify, and guide a missile to the personnel targets at night in rain, fog, and smoke. It could be determined that PGD-65 was the only R & D project necessary to develop product requirement R601. If that were the case, then

R601. ≡ PGD-65.

would be the translation index for R601 and PGD-65. The translation index permits translation of requirements rank-ordered lists into R & D project rank-ordered lists.

B. Computation Inputs

To perform the R & D project prioritization demonstration, several sublists of judge rank orders must be input into the model. The characteristics of the laboratory management environment and the sublists will be described as each is given. The necessary controls for study and for the translations from requirements will be described. Last, the weighting technique and values will be discussed.

1. Sublists Data

The R & D laboratory is faced with multiple sources of suggested rank ordering of its R & D projects. These sources include requirements studies and other documents, headquarters management, and the local laboratory management. The laboratory director, as the formal decision-maker, must evaluate each sublist source and each

alternative to determine what special emphasis should be given to each. The alternatives are considered of equal special emphasis for purposes of this prioritization study. Therefore, each alternative is given an equal alternative weight of one. The judge sublists are given judge weights corresponding to their sources. For example, each directorate manager's sublist has a weight of five, while the laboratory director's sublist has a weight of ten.

a) R & D Projects Sublists

The laboratory director has his own preferences for the rank-ordered priorities of this R & D projects. In an autocractic organization, that list would prevail and this study would be a mute effort. But in a participatively managed organization, the director will choose to give reasonable consideration to the recommendations of his inferior managers as well as those of his superiors.

The alternative R & D projects have been numbered with index values from 1 to 95. Each alternative project has a weight of one. In addition, each project is identified as a member of a technology category. In the prior example for PGD-65, the GD symbols represent the technology category for this project. The laboratory director concentrates his attention on the management, balance, and resources of the technology categories, while the directorate managers concentrate on managing the projects within a category. Therefore, the laboratory director's sublist will be a rank-ordered list of the technology categories. The model will use this ranked category list, with its corresponding projects, to develop a single list of projects for the director. The study will include the option case where he considers the projects as unranked within each category, and the option case

where he accepts the directorate managers recommendations for the project rankings within each category. The laboratory director's technology category preferences are listed in Table 10.

TABLE 10. LABORATORY DIRECTOR'S SUBLIST

Rank	Category Index	Category Indentification
1	1	ES
2	2	GG
3	3	ET
4	5	GD
5	6	DS
6	7	KP
7	4	EHG

Each directorate manager ranked the R & D projects within his category. The sublists for each category manager are listed in Table 11.

TABLE 11. CATEGORY MANAGER'S SUBLIST

Category ES		
Rank	Project Index	Project Indentification
1	1	PES-1
2	2	PES-2
3	3	PES-3
4	4	PES-4
5	5	PES-5
6	6	PES-6
7	7	PES-7
8	8	PES-8
9	9	PES-9
10	10	PES-10
11	11	PES-11
12	12	PES-12
13	13	PES-13
14	14	PES-14
15	15	PES-15
16	16	PES-16

TABLE 11. (CONTINUED)

Category GG

Rank	Project Index	Project Indentification	Rank	Project Index	Project Indentification
1	17	PGG-17	20	36	PGG-36
2	18	PGG-18	21	37	PGG-37
3	19	PGG-19	22	38	PGG-38
4	20	PGG-20	23	39	PGG-39
5	21	PGG-21	24	40	PGG-40
6	22	PGG-22	25	41	PGG-41
7	23	PGG-23	26	42	PGG-42
8	24	PGG-24	27	43	PGG-43
9	25	PGG-25	28	44	PGG-44
10	26	PGG-26			
11	27	PGG-27			
12	28	PGG-28			
13	29	PGG-29			
14	30	PGG-30			
15	31	PGG-31			
16	32	PGG-32			
17	33	PGG-33			
18	34	PGG-34			
19	35	PGG-35			

Category ET

Rank	Project Index	Project Indentification
1	45	PET-45
2	46	PET-46
3	47	PET-47
4	48	PET-48
5	49	PET-49
6	50	PET-50
7	51	PET-51
8	52	PET-52
9	53	PET-53
10	54	PET-54

Category EHG

1	55	PEHG-55
2	56	PEHG-56
3	57	PEHG-57

Category GD

Rank	Project Index	Project Indentification
1	58	PGD-58
2	59	PGD-59
3	60	PGD-60
4	61	PGD-61
5	62	PGD-62
6	63	PGD-63
7	64	PGD-64
8	65	PGD-65
9	66	PGD-66
10	67	PGD-67
11	68	PGD-68
12	69	PGD-69
13	70	PGD-70
14	71	PGD-71

TABLE 11. (CONCLUDED)

Category DS			Category KP		
Rank	Project Index	Project Indentification	Rank	Project Index	Project Indentification
1	72	PDS-72	1	83	PKP-83
2	73	PDS-73	2	84	PKP-84
3	74	PDS-74	3	85	PKP-85
4	75	PDS-75	4	86	PKP-86
5	76	PDS-76	5	87	PKP-87
6	77	PDS-77	6	88	PKP-88
7	78	PDS-78	7	89	PKP-89
8	79	PDS-79	8	90	PKP-90
9	80	PDS-80	9	91	PKP-91
10	81	PDS-81	10	92	PKP-92
11	82	PDS-82	11	93	PKP-93
			12	94	PKP-94
			13	95	PKP-95

The laboratory director's staff, from its more analytical viewpoint than the director's, ranks the projects. The top thirty items sublist for the laboratory staff are listed in Table 12.

TABLE 12. LABORATORY STAFF SUBLIST

Rank	Project Index	Rank	Project Index
1	1	16	95
2	3	17	62
3	4	18	50
4	2	19	47
5	7	20	49
6	45	21	44
7	59	22	32
8	72	23	27
9	80	24	28
10	78	25	24
11	79	26	22
12	83	27	15
13	89	28	10
14	92	29	6
15	94	30	11

Several tri-service committees exist which study restricted areas of DOD technology activities and make recommendations to improve the effectiveness and efficiency of technology advancement efforts of the DOD laboratories. These recommendations also identify and remedy overlap and balance between laboratories. In this demonstration study, the laboratory director has been provided a recommended priority list for his laboratory's projects by the DOD tri-service Terminal Guided Submunitions (TGSM) Committee and the DOD Antitank Guided Munitions (ATGM) Committee. The sublists for the TGSM Committee and the ATGM Committee are listed in Table 13.

TABLE 13. TGSM COMMITTEE SUBLIST

Rank	Project Index	Rank	Project Index	Rank	Project Index
1	2	7	4	13	27
2	45	8	5	14	28
3	49	9	10	15	54
4	51	10	12	16	6
5	46	11	11	17	14
6	47	12	15		

ATGM COMMITTEE SUBLIST

Rank	Project Index	Rank	Project Index
1	2	7	28
2	1	8	12
3	7	9	83
4	47	10	87
5	17	11	95
6	27	12	93

The laboratory director, as the decision-maker, determines the weights to be applied to each judge's sublist. The weights will be applied by weighting type two which is multiplicative. The weights for the judges who prepared R & D project sublists are in Table 14.

TABLE 14. PROJECTS JUDGE'S WEIGHTS

Judge	Weight (WTJ)
Laboratory Director	10
Category ES Manager	3
Category GG Manager	3
Category ET Manager	3
Category EHG Manager	3
Category GD Manager	3
Category DS Manager	3
Category KP Manager	3
Laboratory Staff	5
TGSM Committee	5
ATGM Committee	5

2. Requirements Sublists

Two requirements priority lists are available for the decision maker to use in this R & D projects prioritization study. The alternative product requirements have been numbered with index values from 1 to 44. Each alternative requirement has a weight of one. In addition, each requirement is identified as a number of a requirement category.

One requirements sublist, BDP-ST COM, rank orders all of the requirements based on one categorization of ultimate utilization. The sublist for BDP-ST COM is listed in Table 15.

TABLE 15. BDP-ST COM REQUIREMENTS SUBLIST

Rank	Reqmt. Index	Reqmt. Ident.	Category	Rank	Reqmt. Index	Reqmt. Ident.	Category
1	1	R101.	1	23	23	R401.00I	4
2	2	R106	1	24	24	R401.00K	4
3	3	R108.	1	25	25	R401.00N	4
4	4	R204.	2	26	26	R402	4
5	5	R213.	2	27	27	R404	4
6	6	R301.	3	28	28	R405	4
7	7	R302.00A2	3	29	29	R501.7	5
8	8	R302.00E2	3	30	30	R502.2	5
9	9	R303	3	31	31	R601	6
10	10	R304	3	32	32	R602.	6
11	11	R305.00F	3	33	33	R603	6
12	12	R306	3	34	34	R604	6
13	13	R307	3	35	35	R606	6
14	14	R309	3	36	36	R607	6
15	15	R310.00I	3	37	37	R609	6
16	16	R310.00K	3	38	38	R610	6
17	17	R312.00B	3	39	39	R611	6
18	18	R313.00A	3	40	40	R612	6
19	19	R401.00A	4	41	41	R615	6
20	20	R401.00B	4	42	42	R616	6
21	21	R401.00G	4	43	43	R701.8	7
22	22	R401.00H	4	44	44	R802	8

In the prior example for R601, the 60 integers represent the requirements category. In the requirements sublist, ST, which uses categories, the requirements categories (1 through 8) are unranked and considered equal. The requirements within each category are ranked. The sublists for each requirements category are listed in Table 16.

TABLE 16. REQUIREMENTS CATEGORY SUBLISTS

Category ST1		Category ST2	
Rank	Project Index	Rank	Project Index
1	1	1	4
2	2	2	5
3	3		

Category ST3		Category ST4	
Rank	Project Index	Rank	Project Index
1	6	1	19
2	7	2	20
3	8	3	21
4	9	4	22
5	10	5	23
6	11	6	24
7	12	7	25
8	13	8	26
9	14	9	27
10	15	10	28
11	16		
12	17		
13	18		

TABLE 16. (CONCLUDED)

Category ST5	
Rank	Project Index
1	29
2	30

Category ST6	
Rank	Project Index
1	31
2	32
3	33
4	34
5	35
6	36
7	37
8	38
9	39
10	40
11	41
12	42

Category ST7	
Rank Rank	Project Index
1	43

Category ST8	
Rank	Project Index
1	44

The decision-maker also assigned weight for the requirements judges. The weights are given in Table 17.

TABLE 17. REQUIREMENTS JUDGE'S WEIGHTS

Sublist	Weight (WHT)
BDP-ST COM	10
ST	7

2. Computation Option Controls

a) Options

1) Weights - The multiplicative weighting method (NWT = 2) was chosen for this study with the alternative weights equal to one and the judges' weights input as integers. Though the other methods were considered, the multiplicative was chosen for convenience to the decision-maker.

2) Completion - Most sublist inputs for this study are incomplete since most judges chose to rank the projects or requirements which related to their areas of specialization. The study, using JCONV 3, will consider an alternative case where all sublists are completed by adding all unranked alternatives as equal and lower rank than the list. This will give a comparison with the uncompleted case to determine if any alternatives are moved to other sections of the final priority list.

3) Conversions - The data for this study required conversion for two sublists. The laboratory director's sublist will be one of ranked R & D project categories. The study will examine both the case where the director considered the projects equal in each category and the case where the director accepted the rankings of each category manager.

The second conversion will be that for requirements where the categories are unranked, but the requirements within each category are ranked. In each case above, the model will convert the categories and their alternatives into single prioritized sublists.

b) Translation Equivalences

1) Methodology - Translation of the ranked requirements sublists into equivalent project sublists required several steps. First each requirements sublist was converted into the standard form, but with index integers for requirements.

Using a separate model, the single equivalency statements were input, searched, and compiled in groups by requirements. Over 240 equivalency statements were used in the translation index for this study. Table 18 contains examples of the input translation equivalency statements. Table 19 contains the arranged list of statements stored for access in structuring the translated projects sublist. When the standard requirements sublist was translated, each sublist element was replaced by all of the projects which are shown to be equivalent to that requirement. When multiple projects replaced one requirement, they were considered equal and indifferent. After the substitution step, the translated projects list was purged of all duplicate projects after a project first appeared in the rank order. The resultant standard projects sublist was aggregated like the input projects sublists.

2) Data Sample - Table 18 in the previous section, provides a sample of the data input as equivalency statements. The translation index has approximately 250 statements for this study.

TABLE 18. TYPICAL TRANSLATION EQUIVALENCY STATEMENTS

R301	≡	PES-01
R302.00A2	≡	PES-01
R301	≡	PES-02
R204	≡	PES-02
R401.00A	≡	PES-02
R601	≡	PES-03
R604	≡	PES-03
R204	≡	PES-04
R301	≡	PES-04
R401.00A	≡	PES-04
R301	≡	PES-05
R606	≡	PES-06
R301	≡	PES-07
R606	≡	PES-07

TABLE 19. TYPICAL STORED TRANSLATION STATEMENTS

R204	≡	PES-02
R204	≡	PES-04
R301	≡	PES-01
R301	≡	PES-02
R301	≡	PES-04
R301	≡	PES-05
R301	≡	PES-07
R302.00A2	≡	PES-01
R401.00A	≡	PES-02
R401.00A	≡	PES-04
R601	≡	PES-03
R604	≡	PES-04
R606	≡	PES-06
R606	≡	PES-07

C. Computation Outputs

1. Aggregated Rank Orders

The R & D prioritization study was calculated as cases. In Case 1, the laboratory director considered each technology category project as unranked and the sublists are not synthetically completed. Table 20 presents the final R & D projects ranks rank order for Cases 1 and 2. In Case 2, the laboratory director accepted the technology category projects prioritization by his junior managers. Also, in Case 2, all sublists were synthetically completed.

2. Evaluations

The number of circular trials and coefficients of consistency were computed for Cases 1 and 2. The coefficient of concordance was computed for the Case 2 (completed) input sublists. Table 21 contains all evaluation data for this study.

3. Analysis

The outputs in Tables 20 and 21 show that the conditions differing between these cases, namely synthetic completion, made only minor adjustments to the positions of the alternatives in final rank order. The sublist rank orders were all tested as significantly consistent, but the output PREF Case 2 rank order was 0.05 significantly in concordance.

Except for Projects 55, 56, and 57, the laboratory director's preferences fell in lexigraphic order. The effect of the aggregation in Cases 1 and 2, Table 20, is illustrated by the relatively high ranking of Projects 47, 27, 28, 83, and 95 and the relatively low rankings of Projects 3, 10, 23, and 14 as compared to the lexigraphic order.

TABLE 20. FINAL PREFERENCE RANK ORDERS
R & D PROJECTS STUDY-CASES 1 AND 2

Rank Order	Project Index		Rank Order	Project Index	
	Case 1	Case 2		Case 1	Case 2
1	2	1	23	20	29
2	1	2	24	21	32
3	4	4	25	25	30
4	7	5,7,47	26	26	31
5	47	6	27	29	33
6	5,12	3	28	30	34
7	11,15,27	12,27	29	31	35
8	28	11,28	30	10	36
9	6	9,15	31	33	37
10	9	17	32	34	38
11	13	13	33	35	39
12	17	16	34	36	40,46
13	16	18	35	37	41,83
14	3	19	36	38	44
15	44	20	37	39	42
16	32	22	38	40	43
17	24	21	39	41,83	51
18	22	24,45	40	42,51	48,23
19	45	25	41	43,46,95	8
20	18	49	42	48	95
21	19	26	43	59	59
22	49	10	44	52	52

TABLE 20. (CONCLUDED)

Rank Order	Project Index Case 1	Project Index Case 2	Rank Order	Project Index Case 1	Project Index Case 2
45	8,53	53	68	76	76,94
46	62	58	69	77	77
47	58	62	70	14	72
48	60	60	71	91	14
49	54,61	61	72	78	78
50	63	54	73	72	91
51	64	63	74	79	73
52	65	64	75	73	79
53	66	65	76	74	74
54	67	66	77	87	87
55	23,68	67	78	93	90,92
56	69	68	79	90	93
57	70	69	80	92	55
58	71	70	81	55	56
59	81	71	82	56,57	57
60	89	89	83	88	88
61	82	84	84	--	--
62	80	80	85	--	--
63	84	85			
64	85	86			
65	86,94	81			
66	75	75			
67	50	50,82			

TABLE 21. EVALUATION TESTS RESULTS
R & D PROJECTS STUDY

	Case 1	Case 2
Number of Circular Triads, D	1836	2907.5
Coefficient of Consistency, Zeta	0.949	0.919
Significant to 0.05 Level	Yes	Yes
Significant to 0.01 Level	Yes	Yes
Coefficient of Concordance:		
Mean		624
Sum of Deviations Squared		543,098.
Sum of Ties Factor		500,076.
Coefficient, W_1		0.097
Number of Judges		13
Number of Alternatives		95
Chi-Squared		119.1
Degrees of Freedom		94
W Significant to 0.05 Level		Yes
W Significant to 0.01 Level		No

D. Study Results

1. Specific Problems

Several problems had to be satisfied for this study. First, the dimensions of the model arrays had to be resized to handle the unpurged steps within the requirements translation phase. After purging duplicate projects, the 100 by 100 dimensions were adequate. Also, the alternative index dimensions had to be extended because the requirements element indexes were different and in addition to the project elements. All dimensions could not be freely enlarged because the computer model was approaching the maximum capacity of the computer facility.

A concern of this study, and any future real life studies of this nature, is for the decision-maker to establish an appreciation and feel for the sensitivity of the aggregation results to levels of weighting. For example, is a weight ratio of 10 to 5 appropriate between the director's weight value and that of his staff? Possibly 10 to 7 or even 50 to 5 would more accurately depict the decision-maker's judgment. Table 22, presenting no weight versus weighted results, was used to obtain the appreciation for the weight's sensitivity. For example, the rank positions of Projects 14, 23, 83, and 89 differ noticeably between the weighted and unweighted versions of Case 1, due to relative effects of sublist ranks.

2. Methodology

The methodology used for this R & D projects prioritization study was quite flexible in that it permitted easy changing of the run option controls. The model was effective and efficient in that a full run for this study required less than one minute's computer operating time to obtain thorough results. As is evident in this 95-project

study, aggregation of several long rank orders would be impractical and virtually impossible without a computer model and a large computer.

TABLE 22. EFFECTS OF WEIGHTS R & D PROJECTS
STUDY CASE 1

Rank Order	Project Index: Case 1	Project Index: Case 3 No Wts.	Rank Order	Project Index: Case 1	Project Index: Case 3 No Wts.
1	2	1,2	23	20	46
2	1	4	24	21	44,51
3	4	7	25	25	25
4	7	47	26	26	26,59
5	47	45	27	29	29
6	5,12	27	28	30	30
7	11,15,27	12,28,49	29	31	31
8	28	5	30	10	33,62
9	6	3,6,15	31	33	34
10	9	11	32	34	35
11	13	17	33	35	36
12	17	83	34	36	37
13	16	95	35	37	38
14	3	9	36	38	39
15	44	13,22,24	37	39	40
16	32	16	38	40	41
17	24	32	39	41,83	89
18	22	18	40	42,51	42
19	45	19	41	43,46,95	43
20	18	10	42	48	8
21	19	20	43	59	48
22	49	21	44	52	52

TABLE 22. (CONCLUDED)

Rank Order	Project Index: Case	Project Index: Case 3 No Wts.	Rank Order	Project Index: Case 1	Project Index: Case 3 No Wts.
45	8,53	53	68	76	86
46	62	23	69	77	77
47	58	58	70	14	78
48	60	54	71	91	79
49	54,61	14,60	72	78	91
50	63	61	73	72	73,87
51	64	63	74	79	74
52	65	64	75	73	92,93
53	66	65	76	74	90
54	67	66	77	87	55
55	23,68	67	78	93	56
56	69	68	79	90	57
57	70	69	80	92	88
58	71	70	81	55	--
59	81	50	82	56,57	--
60	89	71	83	88	--
61	82	80	--	--	
62	80	81	--	--	
63	84	82			
64	85	84,94			
65	86,94	75			
66	75	85			
67	50	72,76			

3. Study Conclusions

This prioritization study of R & D projects has demonstrated that the model can be used effectively to aggregate long rank-ordered sublists of R & D projects. It further demonstrated that the model can convert and translate rank-ordered sublists of product requirements into equivalent sublists of R & D projects which are then aggregated with the other sublists of R & D projects.

The study did show that deliberate, subjective, and problem peculiar decisions must be made in preparing the inputs and especially in selecting the options. As previously mentioned, the desired level of weight sensitivity must be established. When categorized data exist, the judge who ranks the categories must determine whether he is indifferent between projects within a category or whether he will accept the within-category project ranking of someone else, such as a junior manager specializing in that category. If judge self-evaluation is not used, the judge who develops a partial sublist must decide whether he is indifferent between the remaining projects. If he is indifferent, he can choose for the model to complete the sublist synthetically. If he is not indifferent between the remaining unranked projects, then he probably should not have the sublist synthetically completed.

CHAPTER VIII. SUMMARY AND RECOMMENDATIONS

A. Concluding Summary

The literature search for this research determined that few articles have been written related to R & D resources management prioritization through an ordinal aggregation process. But there is an extensive body of literature on the more general field of social choice for political science, social science, and economics. This research analyzed this large body of material and presented the literature summary by structuring it into time-phased thrust areas.

Numerous methods have been documented to aggregate rank-ordered sublists. Many of the methods have characteristics that prohibit their use for the objectives of this research. Many of the potentially usable methods employed a majority rule. Only a few pieces of literature compared aggregation methods, and even fewer articles evaluated and selected between methods for specific applications. This research identified seven promising majority-rule methods and classified them for their usefulness in the specific application of multiple judge, no feedback, ordinal prioritization of R & D projects.

A model was developed and coded on a large computer to accomplish the sublist aggregation, weighting, hierarchical conversions, requirements translation, and results evaluations. The coded model has been verified. Validation has been successfully performed against 46 examples from the literature of which 13 are included in this report. The model was then demonstrated for an extensive R & D projects prioritization study.

Fuzzy set rank-order methodology was briefly explored and added to the model for an alternative final aggregation rank ordering. The methodology employed was too insensitive for many of the cases computed. The fuzzy set method would rank many alternatives as indifferent when the other three methods developed preference orders between the same alternatives.

B. Research Accomplishments

In reflection on the research reported in this report and Dobbins [13, 14], several findings and accomplishments are apparent. First, this research demonstrated how a field of knowledge, namely, social choice, possesses applicability to engineering management situations. This phenomenon stimulates the question of which other fields of knowledge are being actively developed in another area that could be of direct benefit to engineering management.

This research demonstrated the practicality and limitations of several majority-rule methods that can be used to aggregate ordinal rank orders. Although extensive theoretical research has strived and generally failed to find aggregation methods that always give intransitive results, for the realistic rank order problems examined, intransitivity was not an impediment.

Specifically, this work has shown that diverse and complex R & D management priority lists can be aggregated into a useful single rank-ordered list. A real life example was studied where 13 sublists with 95 projects and 44 requirements were successfully aggregated.

Finally, this researcher has been rewarded by the sense of accomplishment of progressing through all of the steps of a full research project.

C. Recommendations for Future Investigations

1. Methodology

Through the conduct of this research, certain methodology questions recurred which were interesting, but outside of the specific scope of the research.

Black [4] developed a concept he named the single-peakedness criteria for rank-orders. Theoretical studies have shown single-peakedness to be a condition that can be directly associated with transitive rank orders. But none of the literature located gave specifics on the application of the single-peakedness criteria to rank order aggregation problems. Logic can be developed to evaluate the monotonic characteristics along an order as compared to a reference sequence of the alternatives. What is not obvious is how one can efficiently determine the reference sequence that will allow all sublists to be single-peaked. If a method is not developed, then all sequence combinations of the alternatives must be evaluated before an answer can be given. Single-peakedness methodology is further aggravated by indifferences (ties) or partial sublists.

A promising majority-rule method was attributed to Copeland by Goodwin in Thrall [41]. After intensive library and personal inquiry to Copeland's family, a copy of Copeland's memo has not been obtained. Another recommended topic for further study would be to search further to obtain a copy of Copeland's memo and determine why several papers in the literature stated that it was limited to sublist rank orders with no indifference (ties).

The fuzzy set rank-order methodology which was included in the model frequently gave rankings that were too insensitive to differences in

rank between two or more alternatives. Further study could determine what number of alternatives and number of judges would make fuzzy set rank orders adequately sensitive.

2. Model and Computer Code

Several areas also exist for further research work to improve the modeling and computer coding for the aggregation of rank orders.

The dimensions of the computer code arrays are limited by the computer capacity. With no significant changes in the model, the present 100 x 100 dimension limit could be enlarged to 125 x 125 or perhaps 140 x 140, but little further. The beneficial solution would be a computer code that was not dimension limited. The approach might be to develop a computer code that will progress through very large matrices one section at a time until all sections are computed. This type of modification might permit the model from this research to be used for aggregation of rank-ordered preferences of segments of the population.

The present model, to minimize data storage requirements, does not hold input sublist data as the computations progress through the arrays. This space saving requires that all data be re-input for each problem even if only a single control value changed. Again, extended space capacity could remedy this input data repetition requirement. Further research might find other remedies.

The COMPARE subroutine used Kendall's concordance tests to evaluate pairs of final aggregated rank orders. Kendall's concordance method was necessary where more than two rank orders were evaluated. But there were other methods, such as Kendall's Tau method, that could be considered where there are only two rank orders. An investigation could determine if Kendall's concordance test should be replaced for these final comparative tests.

In conclusion, the research for this report is believed to be a contribution to the field of engineering management knowledge. But as in most research, the development of knowledge has revealed additional questions to be answered.

APPENDICES

Appendix A. FUNCTIONAL FLOW DIAGRAM

The functional flow diagram for the aggregation computer model is presented in Figure A-1. The P term repeated in the flow means PRINT. It concerns the information about the steps in the process.

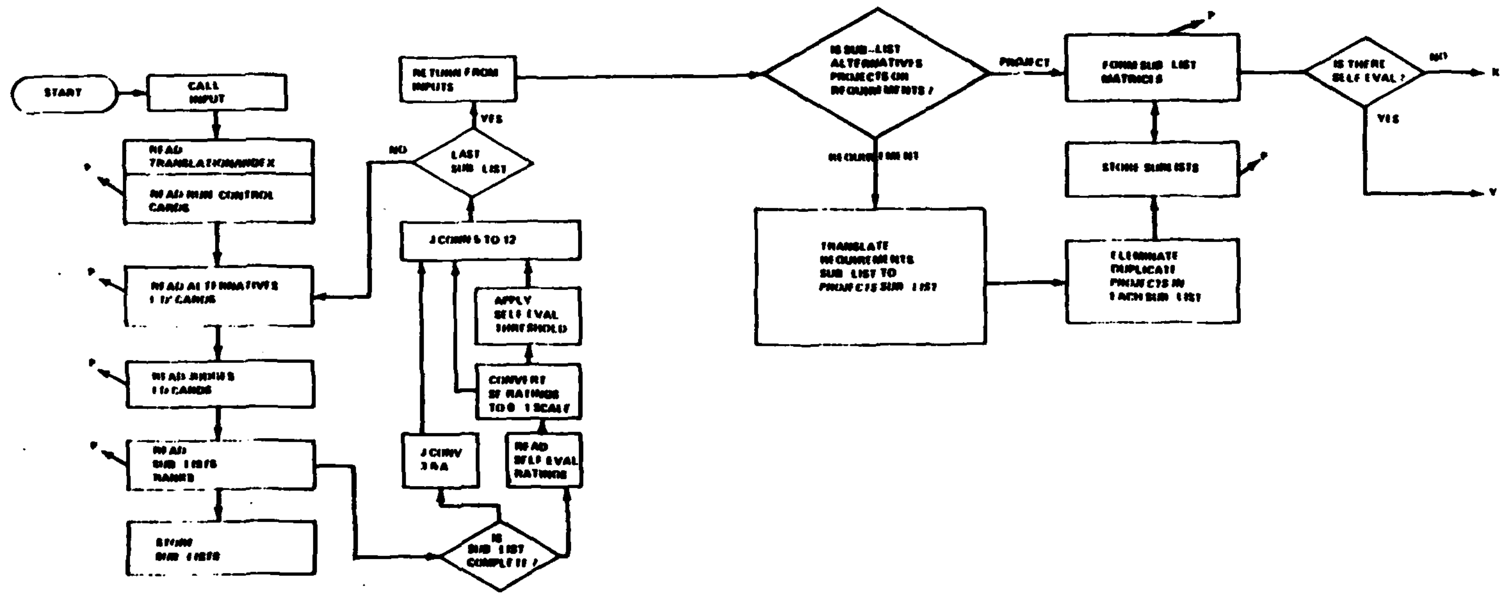


Figure A-1. Model functional flow diagram.

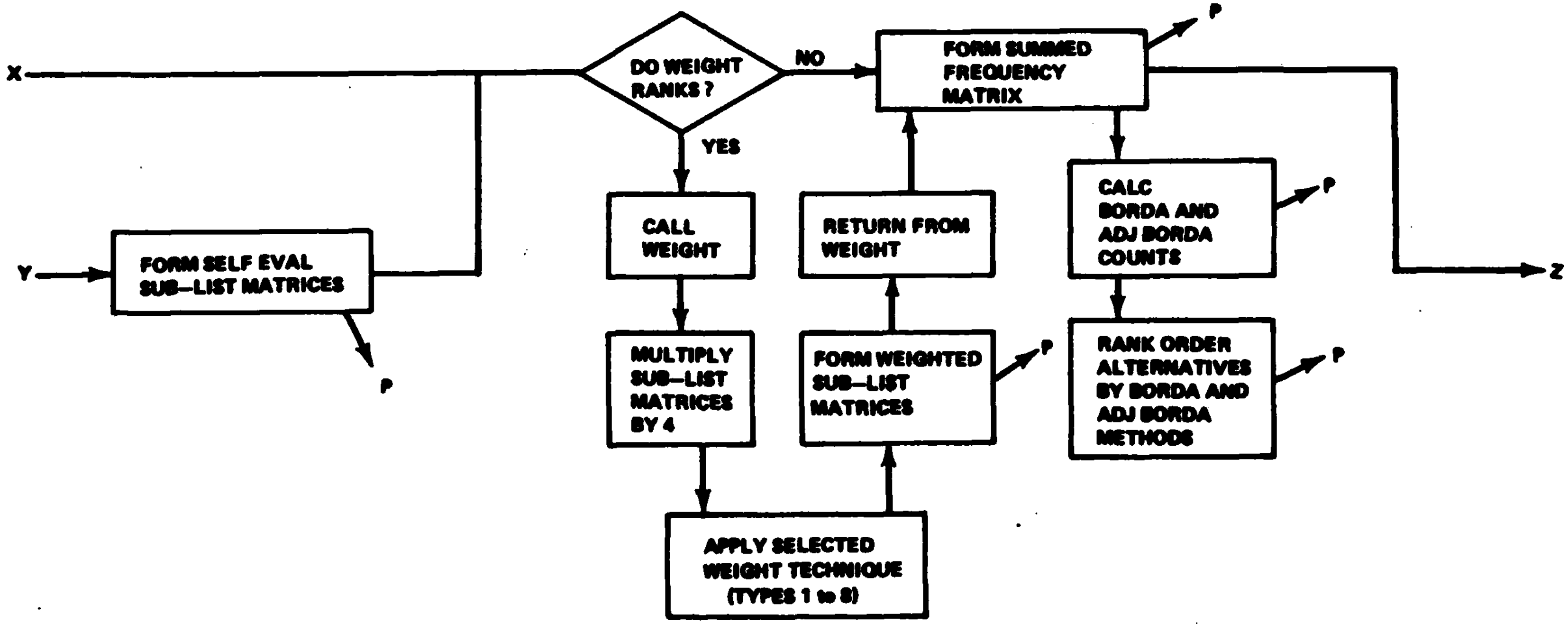


Figure A-1. (Continued).

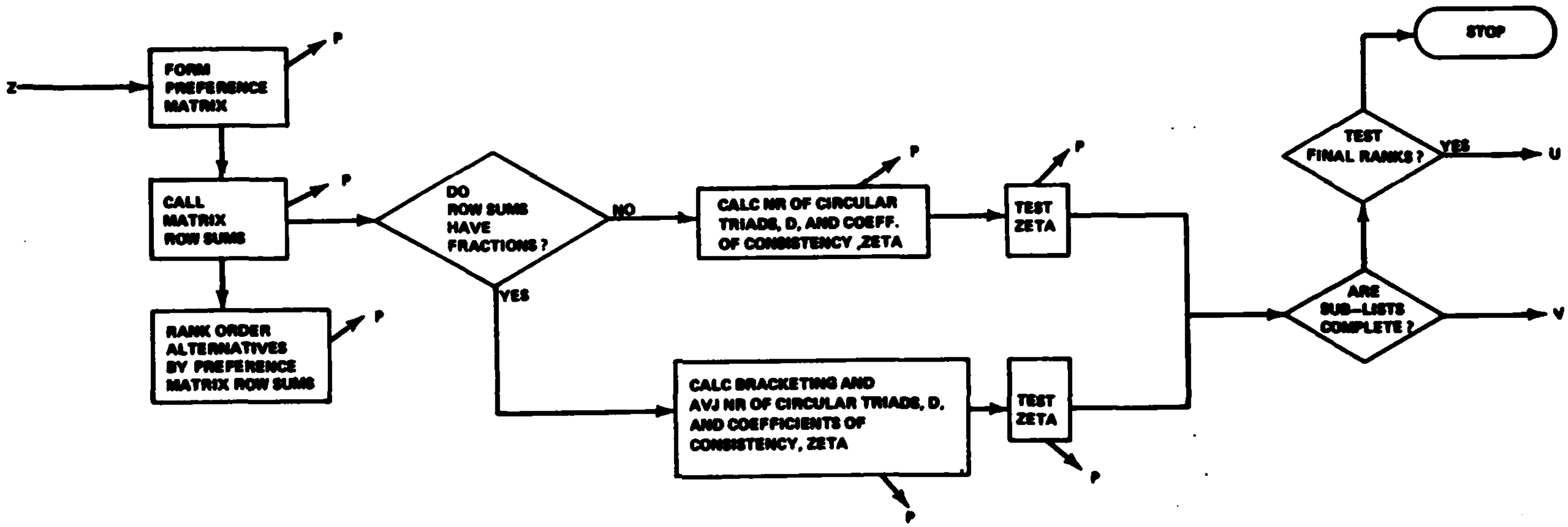


Figure A-1. (Continued).

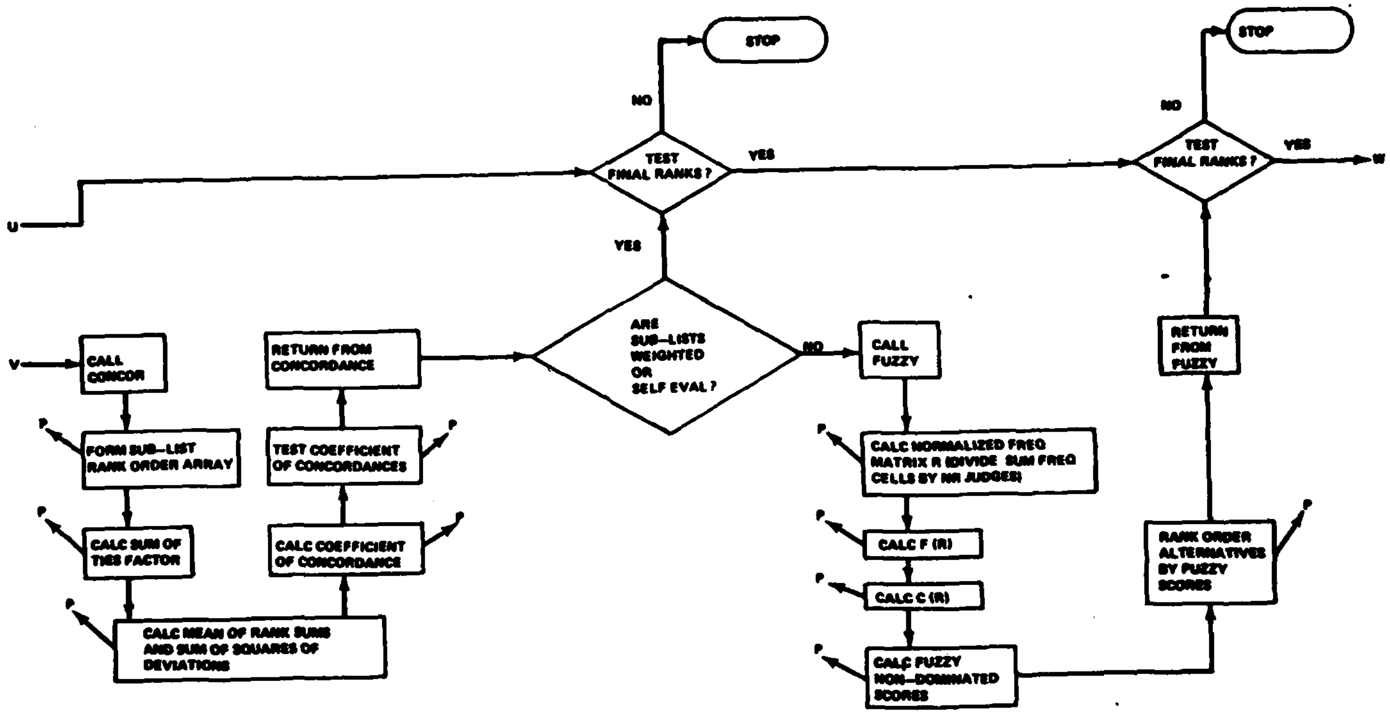


Figure A-1. (Continued).

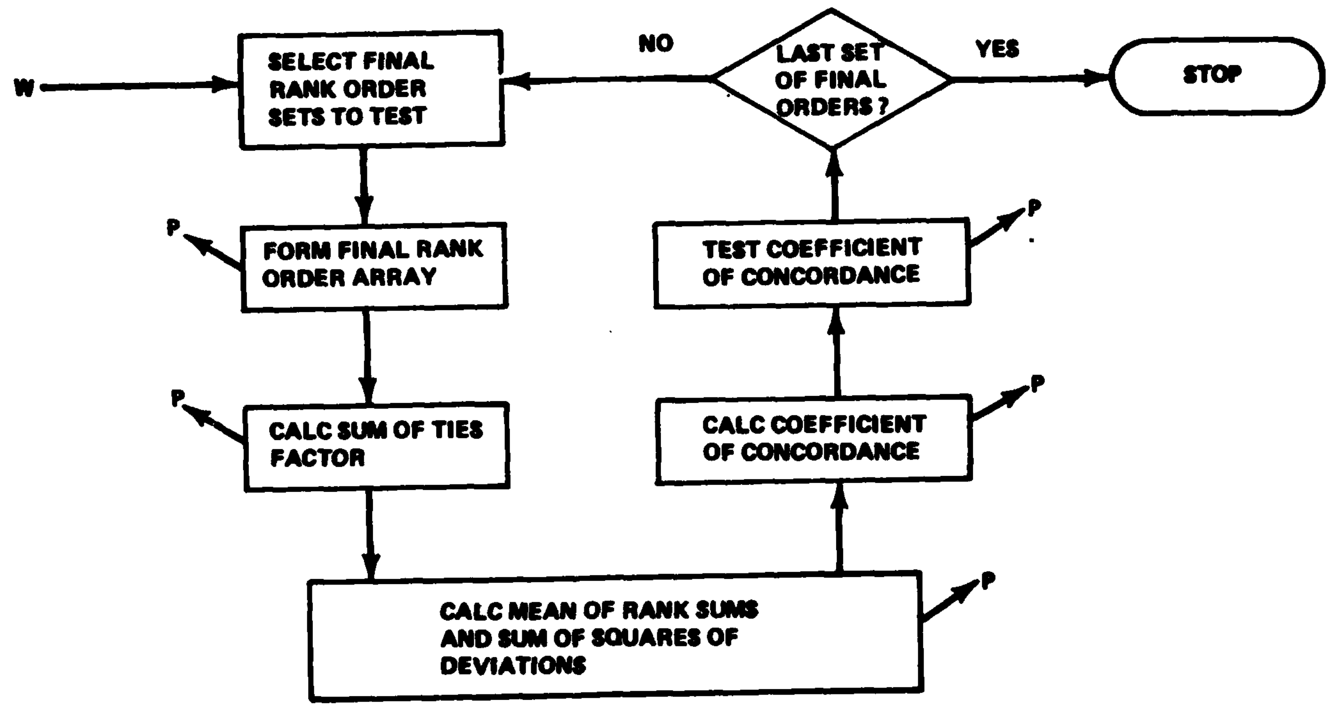


Figure A-1. (Concluded).

Appendix B. INPUT INSTRUCTIONS

Input instructions for leading the controls and data into the model are presented in Table B-1.

TABLE B-1. TECHNOLOGY PLANNING PRIORITIES

Input Requirements

Card Type 1: Header - Name of priority group

Col 01-80

Card Type 2: Control card

Col 05-05 = NWT = Weight type (1-8)

(see Dobbins [14] for descriptions)

Col 10-10 = NCOMP = Complete all matrices if nonzero

Col 15-15 = NPTYP1 = Type of calculation for frequency matrix

0 = 0, .5, 1

1 = -1, 0, 1

Col 20-20 = NPTYP2 = Type of calculation for preference matrix

0 = 0, .5, 1

1 = -1, 0, 1

Col 25-25 = MATR = Self evaluation key

0 = No self evaluation

1 = Self evaluation, complete matrix

2 = Self evaluation, threshhold, reduced matrix

Col 26-30 = THLD = Percentage level under which elements are discarded

Col 35-35 = NPRINT = PRINT control

0 = Print all

1 = No print of sublist frequency matrices

2 = No print of sublist frequency matrices or weighted sublist frequency matrices

3 = Same as NPRINT = 1 plus no print FUZZY

4 = Same as NPRINT = 2 plus no print FUZZY

5 = Print only input and output

6 = No print FUZZY

Card Type 3: Input Type

Col 5 = JELE = Element code

0 = End

1 = Requirements

2 = Projects

Col 10-20 = NELE = Element type name

Card Type 4: 1 - NBR   NBR = Number of requirements

Col 03-05 = K = Element number - Number between 1 - NBR

Col 11-30 = NAM = Element name

Col 31-40 = WHI = Row weight

Col 41-50 = KAT = Category

Terminate Element cards with "END" in Col 11-13

A. Element Number and name are required. If the weight or category factors are blank, they are assumed to be Ø.

B. If a weight type is assigned in Card Type 2, a weight factor must appear on the project card. If the projects are not weighted, but the judges are, then use a one (1) on each card.

Categories are used only in the cases where one or more of the evaluators uses a judge conversion factor of 9, 10, 11, or 12. In which case the CATEGORY (KAT) groups certain projects or requirements together. If the projects within a category are ranked, they must appear in their ranked order.

Element Number	1	2	3	4	5	6	7
Element Name	A	B	C	D	E	F	G
Element Weight							
Element Category	1	1	2	2	2	3	3

The foregoing example implies that 1>2, 3>4>5, 6>7

The final order of the requirements would depend upon the ranked or unranked state of the categories. If, however, the requirements are specified unranked, then the foregoing example would imply

1=2, 3=4=5, 6=7

and again the final order of the requirements would depend upon the ranked or unranked condition of the categories.

Sublist Data Card Sets

Card Type 5: Card 1

Col 01-10 = Judge = Name of judges or office making rank

Col 14-15 = JCONV = 15 Type of project conversion (see Appendix B for descriptions)

Col 16-20 = WTJ = Weight factor of judge.

Col 21-25 = ISEM = 100% weight factor for self evaluation

A. Judge Name - Name of evaluator must be present. If the JCONV or WJT left blank, they are assumed to be Ø.

B. If the JCONV is specified, the program looks for specific data in Card 2 - Free format sublists.

JCONV	Input Requirement
1	Normal input
2	Reduced sublist
3	Input reduced sublist. Program will complete it at end with equal elements all less than the last given element.
4	Input reduced sublist. Program will complete SL at the beginning with equal elements all greater than the first given element.
5	Input rating values in real numbers given in the order of project, e.g., A B C 1, 2, 3, etc. Program will arrange projects in order of highest to lowest, setting equivalent elements equal.

6		Input Julian date of projects in order of projects. Program will arrange projects in order of soonest to latest, setting equivalent elements equal.
7	Input-3	Freeform sublists
	Card A	Key element
	Card B	Secondary array to be inserted into primary array after key element.
	Card C	Primary array - Program inserts secondary array in primary array checking for duplication of each element.
8	Input-3	Freeform sublists
	Card A	Key element
	Card B	Secondary array to be inserted into primary array before key element.
	Card C	Primary array - Program inserts secondary array in primary array checking for duplication of each element.
9-12		Categories must be specified in project cards.
9		Input ranked categories - Categories must not be equal. Program checks for ranking, then groups ranked requirements by category.
10		Input unranded categories - Categories must be equal. Program checks for ranking, then groups ranked requirements by category.
11		Input Ranked Categories - Categories must not be equal. Program checks for ranking, then groups unranked requirements by category.
12		Input unranked categories - Categories must be equal. Program checks for ranking, then groups unranked requirements by category. (If categories are improperly input, an error message is written and the sublist is dropped from calculations.)

C. If weight type factor appears on Card Type 2, a weight factor must appear on the evaluator card. If projects are weighted, but not the judges, then use a one (1) on each card.

Card Type 6: Card 2 - Free format sublist ranks by Project Number. Sequence indicates preference, prefix with minus to indicate equal. Terminate list with a *. Follow special rules for specific JCONV outlined above.

Card Type 7: Self evaluation of expertise in the technical field of each input element. These ratings must be between Ø and ISEM in the element index order. Use Card Type 7 only when MATR = 1 or 2 (Col 25 Card Type 2). This is a free format list of integers terminated with an *.

Appendix C. CODE LISTING

The FORTRAN IV code listing for the model computer code is presented in Table C-1. A more comprehensive description of the computer code, including examples can be found in Dobbins [14], a report of the US Army Missile Laboratory.

TABLE C-1. FORTRAN IV CODE LISTING

```
PROGRAM DOBBINS     74/74    OPT=1                                  FTN 4.6+439

              PROGRAM DOBBINS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE9)
      C
      C          TECHNOLOGY PLANNING PRIORITIES
      C
      C                              JOB 1693  ED DOBBINS / W. JONES
      C
      C
      C       COMMON /CDATA/ NBR ,NJ,NWT, NAME(2,300),A(100,100),WI(300),WJ(101)
             *              , JNAME(101),NSIZE(101),JSUBL(100,101)
              COMMON/HELP/NV,IND(300),ICAT(300),NAM(2),JCHECK(300),ITT,IMAX,JMAX
              COMMON/RANK/LISTC(100,3),LIST(100),LAB(3)
              COMMON/IOO/HEADER(8),NOTCOM,NPTYP1,NPTYP2,NFUZ,NPRINT,JTIE
              COMMON/  /MATR,THLD,SFM(100),ESR(100,101)
      C
              DIMENSION  SUM(200), ADJB(100)
              DIMENSION IPREF(2,100),JPREF(100),JRANK(100)
              DATA NGT/" >"/,  NEQ/" ="/
              DATA  NDASH/"------"/
           10 READ(5,902)  HEADER
          902 FORMAT( 8A10 )
              IF( EOF( 5 )  .NE.  0.0 )  STOP 777
              WRITE(6,900)  HEADER
          900 FORMAT("1TECHNOLOGY PLANNING PRIORITIES",10X,8A10//)
      C
      C                         READ INPUT
              CALL INPUT
      C
      C          COMPUTE FREQUENCY MATRIX FOR EACH SUB-LIST
              CALL  FREQ
      C
      C          PRINT  SUMMED FREQUENCY MATRIX
              IF (NPRINT.GE.3.AND.NPRINT.LT.7)NFUZ=1
              IF (NPRINT.EQ.5) GO TO 505
              IF (NWT.GE.1.AND.NWT.LT.9) GO TO 500
              IF (JTIE.EQ.1) GO TO 450
      C       PRINT SUMMED FREQ MATRIX - JTIE=0, NO WEIGHTS
              WRITE(6,946)  HEADER, ( J , J=1,NBR )
          946 FORMAT ("1SUMMED FREQUENCY MATRIX",20X,8A10,//"   BORDA    ADJ",
             *(T21,18I6))
              WRITE(6,947)   ( NDASH, J=1,NBR )
          947 FORMAT (1X,T20,18A6)
              GO TO 505
          500 IF  (JTIE.EQ.1) GO TO 501
      C       PRINT SUMMED FREQ MATRIX - JTIE=0, W/WEIGHTS
              WRITE (6,949) HEADER,(J,J=1,NBR)
          949 FORMAT  ("1SUMMED FREQUENCY MATRIX",20X,8A10//," EQUIV   EQUIV",
             */," BORDA     ADJ     BORDA    ADJ    ",(T34,15I6))
              WRITE (6,950) (NDASH,J=1,NBR)
          950 FORMAT (1X,T34,15A6)
              GO TO 505
      C       PRINT SUMMED FREQ MATRIX - JTIE=1, W/WEIGHTS
          501 WRITE (6,990) HEADER,(J,J=1,NBR)
          990 FORMAT ("1SUMMED FREQUENCY MATRIX",20X,8A10,//" JUDGE INDIFFERENCE
             * EXISTS",//"   EQUIV",/" ADJ BORDA  ADJ BORDA",(T24,18I6))
              WRITE (6,991) (NDASH,  J=1,NBR)
          991 FORMAT (1X,T24,18A6)
              GO TO 505
```

TABLE C-1. (CONTINUED)


```
      C     PRINT SUMMED FREQ MATRIX - JTIE=1, NO WEIGHTS
        450 WRITE (6,992) HEADER,(J,J=1,NBR)
        992 FORMAT ("1SUMMED FREQUENCY MATRIX",20X,8A10,//" JUDGE INDIFFERENCE
           * EXISTS",//" ADJ BORDA",(T15,18I6))
            WRITE (6,993) (NDASH,J=1,NBR)
        993 FORMAT (1X,T15,18A6)
        505 CONTINUE
            DO 620 I = 1,NBR
      C             BORDA COUNT AND ADJ BORDA
            SUMR = 0.0
            DO 610 J = 1,NBR
        610 SUMR = SUMR + A(I,J)
            SUMC = 0.0
            DO 615 J = 1,NBR
        615 SUMC = SUMC + A(J,I)
            ADJ = SUMR - SUMC
            SUM(I) = SUMR
            ADJR(I) = ADJ
            IF (NPRINT.EQ.5) GO TO 620
            IF (NWT.GE.1.AND.NWT.LT.8) GO TO 600
            GO TO 625
        600 IF (JTIE.EQ.1) GO TO 601
            WRITE (6,951) SUMR,ADJ,SUMR/4,ADJ/4,I, (A(I,J),J=1,NBR)
        951 FORMAT (1X,4F7.1,1X,I3,(T35,"I ",15F6.1))
            GO TO 620
        601 WRITE (6,994) ADJ,ADJ/4.,I,(A(I,J),J=1,NBR)
        994 FORMAT (1X,2(F7.1,1X),I3,(T22,"I ",18F6.1))
            GO TO 620
        625 CONTINUE
            IF (JTIE.EQ.1) GO TO 602
            WRITE(6,968)  SUMR, ADJ, I, ( A(I,J) , J=1,NBR )
        968 FORMAT (1X,2F7.1,1X,I3(T22,"I ",18F6.1))
            GO TO 620
        602 WRITE (6,995) ADJ,I,(A(I,J),J=1,NBR)
        995 FORMAT (1X,F7.1,I3,2X,(T13,"I ",18F6.1))
        620 CONTINUE
            IF (JTIE.EQ.1) GO TO 630
      C
            CALL  ORDER( "BORDA     " , NBR , SUM )
        630 CONTINUE
            CALL  ORDER( "ADJ       " , NBR , ADJR )
      C
      C          COMPUTE PREFERENCE MATRIX
            CALL PREF
      C
      C
      C
      C     BYPASS IF WEIGHTED  AND INCOMPLETE
            IF (NOTCOM.NE.0) GO TO 15
            IF (NFUZ.NE.0) GO TO 9
            CALL FUZZY
          9 CONTINUE
      C
            IF (NPRINT.EQ.5) GO TO 16
            CALL CONCOR
         15 CALL COMPARE
      C
```

TABLE C-1. (CONTINUED)


```
C
C              PRINT PREFERENCE SUMMARY
   16 CONTINUE
      DO 800 J=1,NRR
      K=LISTC(J,2)
      L=IABS(K)
      JRANK(J)=L
      JPREF(J)=NGT
      IF (K.LT.0) JPREF(J)=NEQ
  800 CONTINUE
      JPREF(1)="  "
      DO 810 I=1,NRR
      DO 810 J=1,NRR
      IF (IABS(LISTC(I,2)).EQ.IND(J)) IPREF(1,I)=NAME(1,J)
      IF (IABS(LISTC(I,2)).EQ.IND(J)) IPREF(2,I)=NAME(2,J)
  810 CONTINUE
      WRITE (6,965) HEADER
  965 FORMAT (1H1,T60, 8A10,///T60"PREFERENCE RANK ORDER SUMMARY"///)
      WRITE (6,962) (JPREF(I),JRANK(I),I=1,NRR)
  962 FORMAT("/   ELEMENT INDEX RANK ORDER",/   20(A2,I3))
      WRITE (6,960)
  960 FORMAT   (//T9, "RANK",7X,"PROJECT")
      J=1
      DO 820 I=1,NRR
      IF (I.NE.1.AND.LISTC(I,2).GT.0) J=J+1
      WRITE (6,961) J,IPREF(1,I),IPREF(2,I)
  820 CONTINUE
  961 FORMAT (//T10,I3,5X,2A10)
      GO TO 10
      END
```

CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM
89	I	29 CD 89	SEPARATOR MISSING. SEPARATOR ASSUMED HERE.

TABLE C-1. (CONTINUED)


```
C          INPUT SUBROUTINE
C
C                    KEY VARIABLES
C
C           VARIABLE  AND DISCRIPTION                              SUBROUTINE
C
C          SUM       = "BORDA" COUNT VARIABLE                     DOBBINS
C          ADJR      = "ADJ BORDA" COUNT VARIABLE                 DOBBINS
C          NBR       = NUMBER OF ELEMENTS                         INPUT
C          K         = INDEX OF ELEMENT                           INPUT
C          WHI=WII   = ELEMENT WEIGHT FACTOR (ROW)                INPUT
C          KAT       = CATEGORY -ELEMENT GROUP                    INPUT
C          NWT       = WEIGHT TYPE-OVERALL WEIGHTING              INPUT
C                      FACTOR                                     INPUT
C          NCOMP     = FLAG FOR COMPLETION                        INPUT
C          NPTYP1    = FREQUENCY  MATRIX TYPE CONVERSION          INPUT
C          NPTYP2    = PREFERENCE MATRIX TYPE CONVERSION          INPUT
C          MATR      = JUDGE SELF EVALUATION FLAG                 INPUT
C          THLD      = PERCENT LEVEL UNDER WHICH THE              INPUT
C                      ELEMENTS ARE DISCARDED                     INPUT
C          NPRINT    = PRINT CONTROL FLAG                         INPUT
C          JUDGE     = NAME OF EVALUATOR                          INPUT
C          JCONV     = JUDGE CONVERSION KEY TO ARRANGE            INPUT
C                      DATA INTO STANDARD FORMAT                  INPUT
C          WTJ=WJJ   = JUDGE WEIGHT FACTOR (COLLUM)               INPUT
C          ISFM      = 100 PERCENT WEIGHT FACTOR FOR SELF         INPUT
C                      EVALUATION                                 INPUT
C          JELE      = CODE FOR INPUT TYPE,E.G.,PROJECTS          INPUT
C                      OR REQUIREMENTS                            INPUT
C          ESR       = JUDGE SELF EVALUATION PERCENTAGE           INPUT
C          FVALUE    = FORMATED OUTPUT FROM PRAM                  INPUT
C          C         = KEY ELEMENT FOR JCONV =7 OR 8              INPUT
C          IMAX      = MAXIMUM NUMBER OF ELEMENTS                 INPUT
C          IND       = ARRAY OF INDEXES OF ELEMENTS               INPUT
C          ITT       = TOTAL NUMBER OF PROJECTS                   INPUT
C          JMAX      = MAXIMUM NUMBER OF JUDGES                   INPUT
C          JNAME     = ARRAY OF JUDGES' NAMES                     INPUT
C          JPREF     = ARRAY OF ='S AND >'S FOR EACH RANK         INPUT
C                      ORDER                                      INPUT
C          JRANK     = ARRAY OF RANK ORDER                        INPUT
C          JSUBL     = ARRAY OF ALL JUDGES' RANK ORDER            INPUT
C          NAME      = ARRAY OF ELEMENT NAMES                     INPUT
C          NFUZ      = FLAG TO OMIT FUZZY                         INPUT
C          NJ        = NUMBER OF JUDGES                           INPUT
C          NSIZE     = ARRAY OF THE NUMBER OF ELEMENTS            INPUT
C                      EACH JUDGE RANKED                          INPUT
C          NV        = NUMBER OF ELEMENTS FOUND BY PRAM           INPUT
C                      IN EACH RANK ORDER                         INPUT
C          SEM       = ARRAY OF SELFEV EVALUATION VALUES          INPUT
C                      READ BY PRAM                               INPUT
C          WI        = ARRAY  OF ELEMENT WEIGHTS                  INPUT
C          WJ        = ARRAY  OF JUDGE   WEIGHTS                  INPUT
C          LIST      = ARRAY OF ELEMENTS FOR COMPOSITE RANK       ORDER
C                      ORDER                                      ORDER
C          LISTC     = ARRAY OF ELEMENTS FOR THE THREE            ORDER
C                      COMPOSITE RANK ORDERS                      ORDER
C          LAB       = NAME OF THE COMPOSITE RANK ORDERS          ORDER
```

TABLE C-1. (CONTINUED)

```
                 74/74    OPT=1                              FTN 4.6+439

      C          CS        = CHI-SQUARE STATISTIC                     PREF
      C          D         = KENDALL D =NR OF CIRCULAR TRIADS IN      PREF
      C                      PREF                                     PREF
      C          DS        = VALUES OF KENDALL D AT RANGE LEVELS      PREF
      C          LAB       = LABEL FOR RANGE OF KENDALL  D            PREF
      C          GNU       = DEGREES OF FREEDOM                       PREF
      C          NF        = NUMBER OF FRACTIONAL SUMS                PREF
      C          P         = PROBABILITY THAT RANK IS NOT CONSISTANT  PREF
      C          PTEST     = FIXED CRITICAL VALUE OF P                PREF
      C          ZETA      = COEFFICIENT OF CONSISTANCY               PREF
      C          A(I,J)    = NORMALIZED FREQUENCY MATRIX, R           FUZZY
      C          TRACEC    = SUM OF THE MAJOR DIAGONAL OF A(I,J)      FUZZY
      C                      MATRIX  SQUARED                          FUZZY
      C          TRACE E   = SUM OF THE MAJOR DIAGONAL OF A(I,J)      FUZZY
      C                      MATRIX  * A(I,J)  TRANSPOSED             FUZZY
      C          FR        =  AVERAGE FUZZINESS IN R                  FUZZY
      C          CR        =  AVERAGE CERTAINTY IN R                  FUZZY
      C          S         = SUM OF DEVIATIONS SQUARED                CONCOR
      C          D         = KENDALLS COEFFICIENT OF CONCORDANCE      CONCOR
      C          RBAR      =  MEAN                                    CONCOR
      C          P         = PROBABILITY OF RANK ORDER CONCORDANCE    CONCOR
      C          NELE      = NUMBER OF ENTRIES ON INDEX               REQUIRE
      C
      C***   INPUT DATA    ***
      C
      C** CARDS  1 THRU NBR                    NBR = NUMBER OF PROJECTS
      C      COL 03-05  = K  = ELEMENT NUMBER.  NUMBER BETWEEN 1 AND NBR.
      C          11-30  =  NAM=  ELEMENT NAME.
      C          36-40 = WHI = ROW WEIGHT      F5.0
      C          46-50 = KAT = CATEGORY        I5
      C      TERMINATE PROJECT CARDS WITH "END" IN COLS 11-13.
      C
      C**   CONTROL CARD
      C      COL 05-05 = NWT = WEIGHT TYPE  1 THRU 8
      C                    0 = NO WEIGHTS
      C      COL 10-10 = NCOMP= COMPLETE ALL MATRICES IF NON ZERO
      C      COL 15-15 = NPTYP1 - FREQUENCY MATRIX TYPE CONVERSION
      C      COL 20-20 = NPTYP2 - PREFERENCE MATRIX TYPE CONVERSION
      C                               0 = 0,.5,1
      C                               1 = -1,0,1
      C      COL 25-25 = MATR - SELF EVALUATION KEY
      C                               0 = NO SELF EVALUATION
      C                               1 = SELF EVALUATION COMPLETE MATRIX
      C                               2 = SELFEVALUATION THRESH HOLD REDUCED MATRIX
      C          26-30 = THLD - LEVEL UNDER WHICH THE ELEMENTS ARE DISCARDED
      C       COL 31-35 = NPRINT - NO PRINT KEY
      C                               0 = PRINT ALL
      C                               1 = NO PRINT OF SUB-LIST FREQUENCY MATRICES
      C                               2 = NO PRINT OF SUB-LIST FREQUENCY MATRICES
      C                                   OR WEIGHTED SUB-LIST FREQUENCY MATRICES
      C                               3 = SAME AS NPRINT  = 1 PLUS NO PRINT FUZZY
      C                               4 = SAME AS NPRINT =2 PLUS NO PRINT FUZZY
      C                               5 = PRINT ONLY INPUT AND OUTPUT
      C                               6 = NO PRINT FUZZY
      C
      C** SUB-LIST DATA CARD SETS **
      C      CARD 1
```

TABLE C-1. (CONTINUED)


```
C     COL 01-10 = JUDGE = NAME OF JUDGE OR OFFICE MAKING RANK  A10
C     COL 14-15 = JCONV = TYPE OF PROJECT CONVERSION           I5
C     COL 16-20 = WTJ   = WEIGHT FACTOR OF JUDGE               F5.0
C     COL 21-25 = ISEM - 100 PERCENT WEIGHT FACTOR FOR SELF EVALUATION
C
C     CARD 2 = FREE FORMAT SUB-LIST RANKS BY PROJECT NUMBER.
C              SEQUENCE INDICATES PREFERENCE. PREFEX WITH MINUS TO
C              INDICATE EQUAL. TERMINATE LIST WITH AN *.
C
C   CARD 3 = FREE FORMAT SELF EVALUATION LEVELS. ONE FOR EACH RANKED
C            ITEM IN ORDER. MUST BE LESS THAN OR EQUAL TO ISEM. LIST
C            MUST BE ENDED WITH AN *.
C
      SUBROUTINE  INPUT
C
      COMMON /CDATA/ NBR ,NJ,NWT, NAME(2,300),A(100,100),WI(300),WJ(101)
     *             , JNAME(101),NSIZE(101),JSUBL(100,101)
      COMMON/IDD/HEADER(8),NOTCOM,NPTYP1,NPTYP2,NFUZ,NPRINT,JTIE
      COMMON/HELP/NV,IND(300),ICAT(300),NAM(2),JCHECK(300),ITT,IMAX,JMAX
      COMMON/  /MATR,THLD,SEM(100),ESR(100,101)
      COMMON /WORK/X(100,100),SUMA(300),JRANK(100),JPREF(100)
      DIMENSION SUMB(300)
C
      REAL  FVALUE(100)
      LOGICAL  ERROR,  JERR,  EOF
      DATA     ERROR/.FALSE./ , JERR/.FALSE./
      DATA  NGT/" >"/, NEQ/" ="/
C         CLEAR DATA
      JMAX=100
      IMAX=300
      ERROR = .FALSE.
      N = 0
      NOTCOM = 0
      DO 10 J=1,IMAX
      WI(J)=1.
      IND(J)=0
   10 ICAT(J)=0
      DO 11 J=1,JMAX
      WJ(J)=1.
      DO 11 I=1,JMAX
   11 A(I,J)=0.0
      NJ = 0
      DO 15 K = 1,101
      JNAME(K) = 1H
      NSIZE(K) = 0
      DO 15 J=1,100
      ESR(J,K)=0.
   15 JSUBL(J,K) = 0
C
C            READ CONTROLS
      READ (5,912) NWT,NCOMP,NPTYP1,NPTYP2,MATR,THLD,NPRINT
      WRITE (6,914) NWT,NPTYP1,NPTYP2,MATR,THLD,NPRINT
  912 FORMAT(5I5,F5.2,I5)
  914 FORMAT(/" NWT=",I2,3X,"NPTYP1=",I2,3X,"NPTYP2=",I2,3X,"MATR=",I2,3
     *X,"THLD=",F4.2,3X,"NPRINT="I2/)
      IF( NCOMP .NE. 0 )  HEADER(8) = " COMPLETED"
      IF( NCOMP .NE. 0 )  WRITE(6,915)
```

TABLE C-1. (CONTINUED)


```
  915 FORMAT(/" COMPLETE ALL SUB-LISTS"/)
      IF( NWT .GE. 0  .OR.  NWT .LE. 8 )  GO TO 70
      WRITE(6,918)  NWT
  918 FORMAT("+",T55,"** ERROR **"I3," IS ILLEGAL WEIGHT TYPE")
      NWT = 0
   70 CONTINUE
      ITT=0
      JNJ=0
      INJ=0
      NFUZ=0
C             BYPASS FUZZY IF WEIGHTED
      IF (NWT.NE.0) NFUZ=1
C             END OF WEIGHTS
C
  400 CONTINUE
      READ (5,900) JELE,NELE
      IF (JELE.EQ.0.AND.JNJ.GT.INJ) NOTCOM=1
      IF (JELE.EQ.0) NBR=ITT
      IF (JELE.EQ.0) GO TO 777
  900 FORMAT (I5,5X,A10)
      WRITE (6,901) NELE
  901 FORMAT (/" INPUT READ IN ",A10/)
      WRITE(6,902)
  902 FORMAT (/"  INDEX     ELEMENT NAME",14X,"WT",10X,"CAT"/)
C     READ ELEMENT  INDEX, NAMES, WEIGHTS, CATEGORIES
   17 READ (5,904) K,NAM,WHI,KAT
  904 FORMAT (1X,I4,5X,2A10,5X,F5.0,5X,I5)
      WRITE (6,904) K,NAM,WHI,KAT
      IF( NAM(1) .EQ. "END" .OR.  K .EQ. 999 )  GO TO 50
      IF (JELE.EQ.2) ITT=K
      IF (JELE.EQ.1) K=ITT+K
      IF (K.LE.IMAX) GO TO 20
      WRITE (6,906) IMAX
  906 FORMAT("+",T55,"** ERROR ** INDEX LARGER THAN"I4)
      ERROR = .TRUE.
      GO TO 40
   20 IF( K .GT. 0 )  GO TO 25
      WRITE(6,907)
  907 FORMAT("+",T55,"** ERROR ** INDEX LESS THAN 1")
      ERROR = .TRJE.
      GO TO 40
   25 IF( IND(K) .EQ. 0 )  GO TO 30
      WRITE(6,908)  K, NAME(1,K), NAME(2,K)
  908 FORMAT("+",T55,"** ERROR ** INDEX"I4 ," HAS ALREADY BEEN DEFINED",
     * " AS ",2A10)
      ERROR = .TRUE.
      GO TO 40
C
   30 IND(K) = K
      ICAT(K)=KAT
      NAME(1,K) = NAM(1)
      NAME(2,K) = NAM(2)
      N=MAX0(N,K)
      IF( WHI .NE. 0. )  WI(K) = WHI
      IF( NWT .EQ. 7  )  WI(K) = WHI
   40 GO TO 17
C
```

TABLE C-1. (CONTINUED)


```
   50 NBR = N
      IF (JELE.EQ.1) NBR=NBR-ITT
      DO 60 J=1,NBR
      IF( IND(J) .GT. 0 )  GO TO 60
      WRITE(6,910)  J
  910 FORMAT (" **ERROR** WHAT IS ELEMENT NAME FOR INDEX",I4)
      ERROR = .TRUE.
   60 CONTINUE
C
  120 CONTINUE
C
      WRITE(6,941)   HEADER
  941 FORMAT("1",9A10/)
C                READ SUB-LIST RANKS AND JUDGES
  500 CONTINUE
      READ (5,930) JUDGE, JCONV, WHJ, ISEM
  930 FORMAT (A10,I5,F5.0,I5)
      IF (JUDGE.EQ."END") GO TO 400
      IF (JCONV.EQ.999) GO TO 400
      WRITE (6,932) JUDGE,JCONV,WHJ,ISEM
  932 FORMAT(//"0",A10,4X,"JCONV =",I5,4X,"JUDGE WEIGHT =",F6.1,4X,"USE
     *VALUE LIMIT =",I5)
      IF( WHJ .EQ. 0.  .AND.  NWT .NE. 7 )  WHJ = 1.0
      IF (MATR.NE.0) NFU7=1
C
      DO 502 J=1,NBR
      JRANK(J)  = 0
  502 JCHECK(J) = 0
      JERR = .FALSE.
C
      PRINT 1,NBR,(NJ+1)
    1 FORMAT (/1X,"TOTAL NBR ALT =",I5,5X,"NR THIS JUDGE =",I5)
      IF (JCONV.GT.8) GO TO 800
      IF (JCONV.EQ.7.OR.JCONV.EQ.8) GO TO 700
      CALL PRAN (FVALUE,NV,-NBR)
C                READ FREE FORMAT DATA, END SUB-LIST WITH  *
C                CONVERT TO INTEGER
      IF (JCONV.EQ.5.OR.JCONV.EQ.6) GO TO 600
  565 CONTINUE
      DO 510 J=1,NV
      K = FVALUE(J)
      L = IABS(K)
      IF( L .GT. 0  .AND.  L .LE. NBR )  GO TO 505
      WRITE(6,934)   J, K
  934 FORMAT(" ** ERROR ** ENTRY NUMBER,"I4," HAS ILLIGAL PROJECT/"
     * "PRODUCT OF",I5 )
      ERROR = .TRUE.
      JERR = .TRUE.
  505 IF( JCHECK(L) .EQ. 0 )  GO TO 508
      WRITE(6,933) K
  933 FORMAT(" ** ERROR **",I5," ALREADY RANKED")
      ERROR = .TRUE.
      JERR = .TRUE.
      GO TO 510
C
  508 JRANK(J) = L
      JPREF(J) = NOT
```

TABLE C-1. (CONTINUED)


```
               JCHECK(L) = L
               IF( K .LT. 0 )  JPREF(J) = NEG
           510 CONTINUE
               JPREF(1) = " "
         C
               WRITE(6,942)          ( JPREF(J), JRANK(J), J=1,NV )
           942 FORMAT(/      /(26(A2,I3)//))
         C
         C                CHECK FOR COMPLETE SUB-LIST
               FLAG = 0
               DO 512 J = 1,NBR
           512 IF( JCHECK(J) .EQ. 0 )  FLAG = 1
               IF (FLAG.NE.0) JNJ=JNJ+1
               IF( FLAG .NE. 0 )  WRITE(6,960)
           960 FORMAT (/" SUBLIST IS INCOMPLETE")
         C
               IF( JERR )  GO TO 500
         C         STORE SUB-LIST
               NJ = NJ + 1
               JNAME(NJ) = JUDGE
               NSIZE(NJ) = NV
               WJ(NJ) = WHJ
               DO 515 J = 1,NV
           515 JSUBL(J,NJ) = FVALUE(J)
         C     JCONV=2   SUBLIST NOT COMPLETED
               IF (JCONV.EQ.2) NOTCOM=1
               IF (JCONV.EQ.2) GO TO 540
               IF (JCONV.EQ.3.AND.MATR.NE.0.OR.JCONV.EQ.4.AND.MATR.NE.0)
              *WRITE (6,920)
           920 FORMAT(/" SELF EVALUATION PROHIBITS COMPLETION"/)
               IF (MATR.NE.0) GO TO 534
               IF (JCONV.EQ.3.OR.JCONV.EQ.4) GO TO 516
         C
         C         COMPLETE SUB-LIST
               IF( NCOMP .EQ. 0    )  GO TO 540
           516 CONTINUE
               IF( NV    .GE. NBR )  GO TO 540
         C
               IF (JCONV.EQ.3.OR.JCONV.EQ.4) INJ=INJ+1
               IF (JCONV.NE.4) GO TO 519
         C      JCONV=4
         C      MOVE OVER FOR LEFT INSERT
               DO 518 J=1,NV
               K1=NBR-J+1
               K2=NV-J+1
           518 JSUBL(K1,NJ)=JSUBL(K2,NJ)
               M=0
               DO 514 J=1,NBR
               DO 513 L=K1,NBR
               IF (J.EQ.IABS(JSUBL(L,NJ)))GO TO 514
           513 CONTINUE
               M=M+1
               JSUBL(M,NJ)=-J
           514 CONTINUE
               JSUBL(1,NJ)=IABS(JSUBL(1,NJ))
               NVV=NBR
               GO TO 532
```

```
  519 CONTINUE
C     JCONV=3
C     RIGHT INSERT
      NVV = NV
      DO 530 K=1,NBR
      DO 520 J=1,NV
      IF( K .EQ. IABS( JSUBL(J,NJ) ) )  GO TO 530
  520 CONTINUE
      NVV = NVV + 1
      JSUBL(NVV,NJ) = - K
  530 CONTINUE
      JSUBL(NV+1,NJ) = IABS(  JSUBL(NV+1,NJ)  )
  532 CONTINUE
      NSIZE(NJ) = NVV
      NV = NVV
C
  540 CONTINUE
C     SELF EVALUATION
  534 CONTINUE
      IF (MATR.EQ.0) GO TO 525
      CALL PRAM (SEM,NV,-NBR)
      DO 555 I=1,NV
      IF (SEM(I).GT.ISEM) GO TO 556
  555 CONTINUE
      GO TO 557
  556 WRITE (6,921)
  921 FORMAT (" ***ERROR*** SELF EVALUATION LEVEL GREATER THAN 100 PERCE
     *NT")
      GO TO 525
  557 CONTINUE
      DO 546 I=1,NV
  546 SEM(I)=SEM(I)/FLOAT(ISEM)
      DO 547 I=1,NV
      DO 547 J=1,NV
  547 IF (J.EQ.IABS(JSUBL(I,NJ))) ESR(I,NJ)=SEM(J)
      WRITE (6,952) (ESR(I,NJ),I=1,NV)
  952 FORMAT (/     /,15(F6.2,3X))
C     THRESH HOLD MATRIX REDUCTION SELF EVALUATION
      IF (MATR.EQ.1) GO TO 525
      DO 548 I=1,NBR
      IF (ESR(I,NJ).LT.THLD) GO TO 549
      GO TO 548
  549 IF (JSUBL(I,NJ).GT.0.AND.JSUBL(I+1,NJ).LT.0) JSUBL(I+1,NJ)=IABS(JS
     *UBL(I+1,NJ))
      JSUBL(I,NJ)=0
  548 CONTINUE
      M=0
      DO 553 I=1,NBR
  553 IF (JSUBL(I,NJ).NE.0) M=M+1
  554 K=0
      DO 550 I=1,NBR
      IF (JSUBL(I,NJ).EQ.0) GO TO 551
      GO TO 550
  551 IF (I.GT.M) GO TO 550
      DO 552 J=I,NBR
      K=1
  552 JSUBL(J,NJ)=JSUBL(J+1,NJ)
```

```
      550 CONTINUE
          IF (K.EQ.1) GO TO 554
          IF (M.NE.NBR) NV=M
          NSIZE(NJ)=NV
          IF (M.NE.NBR) NCOMP=1
      525 CONTINUE
C         REQUIREMENTS TO PROJECTS TRANSLATION
          IF (JELE.EQ.1) GO TO 1050
          GO TO 500
     1050 CALL REQUIRE
          WRITE (6,951)
      951 FORMAT (//1X,"REQUIREMENTS TO PROJECTS TRANSLATION")
          DO 1000 I=1,NV
     1000 JRANK(I)=0
          DO 1010 J=1,NV
          K=JSUBL(J,NJ)
          L=IABS(K)
          JRANK(J)=L
          JPREF(J)=NGT
          IF (K.LT.0) JPREF(J)=NEQ
     1010 CONTINUE
          JPREF(1)="   "
          WRITE (6,950) (JPREF(J),JRANK(J),J=1,NV)
      950 FORMAT (/       /(26(A2,I3)//))
C         GO READ MORE DATA
          GO TO 500
      600 CONTINUE
          NVV=NV-1
          DO 601 I=1,NV
      601 SUMA(I)=FLOAT(I)
C         JCONV=5
C         CONVERSION OF DATA TO DECENDING ORDER
          IF (JCONV.EQ.5) GO TO 625
          DO 605 I=1,NVV
          M=I+1
          DO 605 J=M,NV
          IF (FVALUE(J).LT.FVALUE(I)) GO TO 605
          HOLD=FVALUE(J)
          SHOLD=SUMA(J)
          FVALUE(J)=FVALUE(I)
          SUMA(J)=SUMA(I)
          FVALUE(I)=HOLD
          SUMA(I)=SHOLD
      605 CONTINUE
          M=0
          DO 603 I=1,NV
          IF (FVALUE(I).EQ.0.) SUMA(I)=0.
          IF (SUMA(I).NE.0.)  M=M+1
      603 CONTINUE
          MM=M-1
          DO 602 I=1,MM
      602 IF (FVALUE(I).EQ.FVALUE(I+1)) SUMA(I+1)=-SUMA(I+1)
          GO TO 650
C         JCONV=6
C         CONVERSION OF DATA TO ASCENDING ORDER
      625 DO 610 I=1,NVV
          M=I+1
```

```
      DO 610 J=M,NV
      IF (FVALUE(J).GT.FVALUE(I)) GO TO 610
      HOLD=FVALUE(J)
      SHOLD=SUMA(J)
      FVALUE(J)=FVALUE(I)
      SUMA(J)=SUMA(I)
      FVALUE(I)=HOLD
      SUMA(I)=SHOLD
  610 CONTINUE
      M=0
      DO 612 I=1,NV
      IF (FVALUE(I).EQ.0.) SUMA(I)=0.
      IF (SUMA(I).NE.0.)  M=M+1
  612 CONTINUE
      MM=M-1
      DO 611 I=1,MM
  611 IF (FVALUE(I).EQ.FVALUE(I+1)) SUMA(I+1)=-SUMA(I+1)
  650  IF (M.NE.NV) NV=M
      DO 651 J=1,NV
  651 FVALUE(J)=SUMA(J)
      GO TO 545
  700 CONTINUE
      CALL PRAM (C,NV,1)
      CALL PRAM(SUMB,NVV,-NBR)
      CALL PRAM (SUMA,NV,-NBR)
      IF (JCONV.NE.7) GO TO 750
C     JCONV=7   INSERT AFTER KEY REQUIREMENT
      DO 701 I=1,NV
  701 IF (SUMA(I).EQ. C) K=I
      DO 705 I=1,NVV
      DO 705 J=I,K
  705 IF(SUMA(J).EQ.SUMB(I)) SUMB(I)=0.
      DO 702 I=1,K
  702 FVALUE(I)=SUMA(I)
      NBB=NVV+K
      DO 703 I=1,NVV
  703 FVALUE(I+K)=SUMB(I)
      NVT=NV+NVV
      K1=K+1
      DO 706 I=1,NBB
      DO 706 J=K1,NV
  706 IF (FVALUE(I).EQ.SUMA(J)) SUMA(J)=0.
      DO 704 I=K1,NV
  704 FVALUE(I+NVV)=SUMA(I)
      DO 707 I=1,NVT
      IF (FVALUE(I).EQ.0.) GO TO 710
      GO TO 707
  710 DO 711 J=I,NVT
  711 FVALUE(J)=FVALUE(J+1)
  707 CONTINUE
      M=0
      DO 715 I=1,NVT
  715 IF (FVALUE(I).NE.0.) M=M+1
      NV=M
      GO TO 545
  750 CONTINUE
C     JCONV=8   INSERT BEFORE KEY REQUIREMENT
```

```
      DO 751 I=1,NV
  751 IF (SUMA(I).EQ. C) K=I
      K1=K-1
      DO 755 I=1,NVV
      DO 755 J=I,K1
  755 IF (SUMA(J).EQ.SUMB(I)) SUMB(I)=0.
      DO 752 I=1,K1
  752 FVALUE(I)=SUMA(I)
      NBR=NVV+K1
      DO 753 I=1,NVV
  753 FVALUE(I+K1)=SUMB(I)
      NVT=NV+NVV
      DO 756 I=1,NBR
      DO 756 J=K,NV
  756 IF (FVALUE(I).EQ.SUMA(J)) SUMA(J)=0.
      DO 754 I=K,NV
  754 FVALUE (I+NVV)=SUMA(I)
      DO 757 I=1,NVT
      IF (FVALUE(I).EQ.0.) GO TO 760
      GO TO 757
  760 DO 761 J=I,NVT
  761 FVALUE(J)=FVALUE(J+1)
  757 CONTINUE
      M=0
      DO 765 I=1,NVT
  765 IF (FVALUE(I).NE.0.) M=M+1
      NV=M
      GO TO 545
  800 CONTINUE
      CALL PRAM (FVALUE,NV,-NBR)
      WRITE (6,806) (FVALUE(I),I=1,NV)
  806 FORMAT (/1X,"CATEGORY RANK",/5X,15(F6.0,2X)/)
      IF (JCONV.EQ.10)GO TO 825
      IF (JCONV.EQ.11) GO TO 850
      IF (JCONV.EQ.12) GO TO 875
C     JCONV=9
C     RANKED REQUIREMENTS IN RANKED CATEGORIES
      DO 820 I=2,NV
  820 IF (FVALUE(I).LE.0.) GO TO 821
      M=1
      DO 801 I=1,NV
      DO 802 J=1,NBR
      IF (FVALUE(I).NE.ICAT(J)) GO TO 802
      SUMA(M)=FLOAT(IND(J))
      IF (M.GT.NBR) GO TO 802
      M=M+1
  802 CONTINUE
  801 CONTINUE
      DO 803 I=1,NBR
  803 FVALUE(I)=SUMA(I)
      NV=NBR
      GO TO 545
  821 WRITE (6,823)
  823 FORMAT (/" CATEGORIES NOT RANKED,WRONG JCONV"/)
      GO TO 540
  825 CONTINUE
C     JCONV=10
```

TABLE C-1. (CONTINUED)


```
C      RANKED REQUIREMENTS IN UNRANKED CATEGORIES
       DO 840 I=2,NV
   840 IF (FVALUE(I).GT.0.) GO TO 845
       IF (JELE.EQ.1) GO TO 841
       DO 826 I=1,NV
       SUMB(I)=0.
       DO 826 J=1,NBR
       IF (I.EQ.ICAT(J)) SUMB(I)=SUMB(I)+1.
   826 CONTINUE
       GO TO 843
   841 IF=ITT+1
       IL=NBR+ITT
       DO 842 I=1,NV
       SUMB(I)=0.
       DO 842 J=IF,IL
       IF (I.EQ.ICAT(J)) SUMB(I)=SUMB(I)+1.
   842 CONTINUE
   843 CONTINUE
       M=1
       DO 827 I=1,NBR
       SUM=0.
       DO 827 J=1,NV
       SUMA(M)=FLOAT(I)+SUM
       IF (SUMA(M).GT.NBR) GO TO 827
       IF (M.EQ.1) GO TO 829
       MM=M-1
       DO 828 K=1,MM
       IF (ABS(SUMA(M)).EQ.ABS(SUMA(K)).AND.SUM.EQ.0.)  KK=1
   828  IF ( ABS(SUMA(M)).EQ. ABS(SUMA(K))) GO TO 832
       IF (SUM.NE.0.) SUMA(M)=-SUMA(M)
   829 M=M+1
       IF (KK.EQ.1) SUMA(M-1)=ABS(SUMA(M-1))
       KK=0
   832 SUM=SUM+SUMB(J)
   827 CONTINUE
       DO 835 I=1,NBR
   835 FVALUE(I)   =SUMA(I)
       NV=NBR
       GO TO 545
   845 WRITE (6,846)
   846 FORMAT (/" CATEGORIES NOT UNRANKED, WRONG JCONV"/)
       GO TO 540
   850 CONTINUE
C      JCONV=11
C      UNRANKED REQUIREMENTS IN RANKED CATEGORIES
       DO 855 I=2,NV
   855 IF (FVALUE(I).LE.0.) GO TO 856
       M=1
       DO 851 I=1,NV
       DO 851 J=1,NBR
       IF (FVALUE(I).NE.ICAT(J)) GO TO 851
       SUMA(M)=FLOAT(IND(J))
       IF (FVALUE(I).EQ.ICAT(J-1)) SUMA(M)=-SUMA(M)
       IF (M.EQ.NBR) GO TO 851
       M=M+1
   851 CONTINUE
       DO 852 I=1,NBR
```

TABLE C-1. (CONTINUED)


```
          852 FVALUE(I)=SUMA(I)
              NV=NBR
              GO TO 545
          856 WRITE (6,823)
              GO TO 540
          875 CONTINUE
C             JCONV=12
C             UNRANKED REQUIREMENTS IN UNRANKED CATEGORIES
              DO 876 I=2,NV
          876 IF (FVALUE(I).GT.0.) GO TO 895
              DO 877 I=2,NBR
          877 IND(I)=-IND(I)
              DO 878 I=1,NBR
          878 FVALUE(I)=FLOAT(IND(I))
              NV=NBR
              GO TO 545
          895 WRITE(6,896)
          896 FORMAT (/" CATEGORIES NOT UNRANKED, WRONG JCONV"/)
              GO TO 540
C
          777 CONTINUE
              RETURN
              END
```

TABLE C-1. (CONTINUED)


```
      C         COMPUTE FREQUENCY MATRIX FOR EACH SUB-LIST
      C
            SUBROUTINE  FREQ
      C
            COMMON /CDATA/ NBR ,NJ,NWT, NAME(2,300),A(100,100),WI(300),WJ(101)
           *              , JNAME(101),NSIZE(101),JSUBL(100,101)
            COMMON/IDD/HEADER(8),NOTCOM,NPTYP1,NPTYP2,NFUZ,NPRINT,JTIE
            COMMON /WORK/X(100,100),SUMA(300),JRANK(100),JPREF(100)
            COMMON/  /MATR,THLD,SEM(100),ESR(100,101)
            DATA NDASH/"------"/
            DATA  NGT/" >"/, NEQ/" ="/
      C
            DO 600  K = 1,NJ
            NV = NSIZE(K)
            JUDGE = JNAME(K)
      C
            DO 502 J=1,NBR
        502 JRANK(J)  = 0
      C
            DO 510 J=1,NV
            I = JSUBL(J,K)
            L = IABS(I)
            JRANK(J) = L
            JPREF(J) = NGT
            IF( I .LT. 0 )  JPREF(J) = NEQ
        510 CONTINUE
            JPREF(1) = "  "
      C
      C            SUB-LIST FREQUENCY MATRIX
            CALL MATRIX (NBR,NV,JUDGE,JRANK,JPREF,X)
      C
      C         WEIGHT FREQUENCY MATRIX
            IF (MATR.EQ.0) GO TO 500
      C
      C     SELF EVALUATION WEIGHTING
            DO 515 I=1,NBR
            DO 515 J=1,NBR
            IF (NWT.EQ.0) X(I,J)=4.*X(I,J)
        515 X(I,J)=X(I,J)*ESR(I,K)
            IF (NPRINT.EQ.2.OR.NPRINT.EQ.4.OR.NPRINT.EQ.5) GO TO 500
            WRITE (6,900) HEADER,JUDGE,(JPREF(J),JRANK(J),J=1,NBR)
        900 FORMAT("1SUB-LIST SELF EVALUATION FREQUENCY MATRIX",10X,8A10,//1X,
           *A10,(T12,24(A2,I3)/))
            IF (MATR.EQ.2) WRITE(6,910)
        910 FORMAT (/" REDUCED MATRIX"/)
            WRITE (6,901) (J,J=1,NBR)
        901 FORMAT (/" PROJ",(T7,20I6))
            WRITE (6,902) (NDASH,J=1,NBR)
        902 FORMAT (1X,T7,20A6)
            DO 903 I=1,NBR
        903 WRITE (6,904) I,(X(I,J),J=1,NBR)
        904 FORMAT (1X,I3,(T6,"I ",20F6.2))
        500 CONTINUE
      C
      C     FUNCTIONAL WEIGHTING
            CALL WEIGHT (NWT,NBR,WI,WJ(K),X,JNAME(K))
      C
```

TABLE C-1. (CONTINUED)

```
SUBROUTINE FREQ        74/74    OPT=1                              FTN 4.6+439

C        SUM SUB-LIST
      DO 520 J = 1,NBR
      DO 520 I = 1,NBR
  520 A(I,J) = A(I,J) + X(I,J)
C
C
  600 CONTINUE
C
      RETURN
      END
```

TABLE C-1. (CONTINUED)


```
      C                FREQUENCY MATRIX FOR EACH SUB-LIST
      C
            SUBROUTINE MATRIX (NBR,NV,JUDGE,JRANK,JPREF,X)
      C
      C     NTYPE = 0 TO CONSIDER RANKED VARIABLES ONLY
      C           = 1 TO CONSIDER ALL UNRANKED VARIABLES EQUAL.
      C     NBR   = TOTAL NUMBER OF VARIABLES IN STUDY.
      C     JUDGE = NAME OF SUB-LIST.
      C     JRANK = PROJECTS RANKED IN ORDER OF PREFERENCE.
      C     JPREF = PREFERENCE OF RANKED PROJECTS ( > , = ).
      C     X     = FREQUENCY MATRIX FOR SUB-LIST.
      C     NV    = NUMBER OF RANKED VARIABLES IN SUB-LIST.
      C
            COMMON/IDD/HEADER(8),NOTCOM,NPTYP1,NPTYP2,NFUZ,NPRINT,JTIE
            DIMENSION  X(100,100) , JRANK(100) , JPREF(100)
      C
            DIMENSION  JR(100), JP(100)
            DATA  NGT/" >"/, NEQ/" ="/
            DATA NDASH/"------"/
      C     SELECT TYPE OF CALCULATION
            EQVAL=0.5
            ADVAL=1.0
            IF (NPTYP1.EQ.0) GO TO 5
            EQVAL=0.0
            ADVAL=0.0
          5 CONTINUE
      C
            NVV = NV
            DO 10 J=1,NVV
            JR(J) = JRANK(J)
         10 JP(J) = JPREF(J)
            DO 12 J = 1,NBR
            DO 12 I = 1,NBR
         12 X(I,J) = 0.0
            NVM1=NVV-1
            DO 60 K=1,NVM1
            I = JR(K)
            XX = 1.0
            IF (JP(K+1).EQ.NEQ) XX=EQVAL
            N = K + 1
            DO 40 M=N,NVV
            J = JR(M)
            X(I,J) = XX
            X(J,I)=ADVAL-XX
            IF( JP(M+1) .NE. NEQ )  XX = 1.0
         40 CONTINUE
         60 CONTINUE
      C
            IF (NPTYP1.EQ.1) NFUZ=1
            IF (NPRINT.GE.1.AND.NPRINT.LE.5) GO TO 99
            DO 14 I=1,NBR
            DO 14 J=1,NBR
            IF (NPTYP1.EQ.0.AND.X(I,J).EQ..5) JTIE=1
            IF (NPTYP1.EQ.1.AND.X(I,J).EQ.0.) JTIE=1
         14 CONTINUE
            WRITE(6,902)  HEADER, JUDGE, ( JP(J), JR(J), J=1,NVV )
        902 FORMAT("1SUB-LIST FREQUENCY MATRIX",20X,8A10//1X,A10,(T12,24(A2,I3
```

TABLE C-1. (CONTINUED)


```
     * )/))
      WRITE(6,904)   ( J, J=1,NBR )
  904 FORMAT(/" PROJ",( T7,20I6 ) )
      WRITE(6,905) ( NDASH, J=1,NBR )
  905 FORMAT(1X,T7,20A6)
C
      DO 70 I=1,NBR
      WRITE(6,906)   I, ( X(I,J), J=1, NBR )
  906 FORMAT (1X,I3,(T6,"I ",20F6.1))
   70 CONTINUE
C
   99 CONTINUE
      RETURN
      END
```

TABLE C-1. (CONTINUED)


```
      C          WEIGHT FREQUENCY MATRIX
      C
            SUBROUTINE WEIGHT (NWT,NBR,WI,WJ,X,JNAME)
      C
      C          NWT = WEIGHTING TYPE
      C          NBR = NUMBER OF VARIABLES
      C          WI  = ROW WEIGHT
      C          WJ  = JUDGE WEIGHT
      C          X   = SUB-LIST FREQUENCY MATRIX
      C
            COMMON/IDD/HEADER(8),NOTCOM,NPTYP1,NPTYP2,NFUZ,NPRINT,JTIE
            DIMENSION  WI(100),          X(100,100)
            DATA NDASH/"------"/
      C
            IF( NWT .LT. 1  .OR.  NWT .GT. 8 )  RETURN
            WJJ = WJ
      C
      C              MULT MATRIX BY 4 BEFORE WEIGHTING
            DO 90 I=1,NBR
            DO 90J=1,NBR
         90 X(I,J)=4.*X(I,J)
      C
            GO TO ( 100,200,300,400,500,600,700,800 )  NWT
      C
        100 DO 120 I = 1,NBR
            WII = WI(I)
            DO 120 J = 1,NBR
        120 X(I,J) = WII * X(I,J)
            GO TO 900
      C
        200 DO 220 J = 1,NBR
            DO 220 I = 1,NBR
        220 X(I,J) = WJJ * WI(I) * X(I,J)
            GO TO 900
      C
        300 DO 320 I = 1,NBR
            WII = WI(I)
            DO 320 J = 1,NBR
            IF( X(I,J) .LE. 0.0 )  GO TO 320
            X(I,J) = X(I,J) ** WII
        320 CONTINUE
            GO TO 900
      C
        400 DO 420 J = 1,NBR
            DO 420 I = 1,NBR
            IF( X(I,J) .LE. 0.0 )  GO TO 420
            X(I,J) = X(I,J) ** ( WI(I) * WJJ )
        420 CONTINUE
            GO TO 900
      C
        500 DO 520 J=1,NBR
            DO 520 I = 1,NBR
            IF( X(I,J) .LE. 0.0 )  GO TO 520
            X(I,J) = WI(I) * ( X(I,J) ** WJJ )
        520 CONTINUE
            GO TO 900
      C
```

TABLE C-1. (CONTINUED)


```
      600 DO 620 J = 1,NBR
          DO 620 I = 1,NBR
          IF( X(I,J) .LE. 0.0 )  GO TO 620
          X(I,J) = WJJ * ( X(I,J) ** WI(I) )
      620 CONTINUE
          GO TO 900
C
      700 DO 720 J=1,NBR
          DO 720 I = 1,NBR
          IF( X(I,J) .EQ. 0.0 )  GO TO 720
          X(I,J) = X(I,J) + WI(I) + WJJ
      720 CONTINUE
          GO TO 900
C
      800 DO 820 J = 1,NBR
          DO 820 I = 1,NBR
          IF( X(I,J) .LE. 0 ) GO TO 820
          X(I,J) = ALOG( WI(I) * WJJ * X(I,J) )
      820 CONTINUE
C
C           PRINT WEIGHTED MATRIX
      900 CONTINUE
          IF (NPRINT.EQ.2.OR.NPRINT.EQ.4.OR.NPRINT.EQ.5) RETURN
          WRITE (6,902) HEADER,JNAME,(J,J=1,NBR)
      902 FORMAT(///"1WEIGHTED SUB-LIST FREQUENCY MATRIX",10X,8A10//
         *2X,A10,//" PROJ",(T7,20I6))
          WRITE(6,905) ( NDASH, J=1,NBR )
      905 FORMAT(1X,T7,20A6)
          DO  920 I=1,NBR
          WRITE(6,906)  I, ( X(I,J), J=1,NBR )
      906 FORMAT (1X,I3,(T6,"I ",20F6.1))
      920 CONTINUE
          RETURN
          END
```

TABLE C-1. (CONTINUED)

```
SUBROUTINE ORDER      74/74    OPT=1                              FTN 4.6+439

          C                  PRINT  RANK  ORDER
          C
                SUBROUTINE  ORDER( NAME , NBR , SUMA )
                COMMON/RANK/LISTC(100,3),LIST(100),LAB(3)
                COMMON/IDD/HEADER(8),NOTCON,NPTYP1,NPTYP2,NFUZ,NPRINT,JTIE
          C
                DIMENSION  SUMA( 1 )
                DIMENSION  JRANK(100), JPREF(100)
          C              SORT SUMA
          C
                DO 200 J=1,NBR
                JPREF(J) = " >"
            200 JRANK(J) = J
                JPREF(1) = "  "
          C
                DO 210 J=1,NBR
                DO 210 K=J,NBR
                IF( SUMA(J) .GE. SUMA(K) )  GO TO 210
                TEMP = SUMA(J)
                JP   = JRANK(J)
                SUMA(J) = SUMA(K)
                JRANK(J) = JRANK(K)
                SUMA(K) = TEMP
                JRANK(K) = JP
            210 CONTINUE
          C
                IF (NPRINT.EQ.5.AND.NAME.NE."PREF") RETURN
                IF( NAME .EQ. " " )  RETURN
          C     STORE FOR CONCORDANCE
                IF (NAME.EQ."FUZZY") LAB(3)=NAME
                IF (JTIE.EQ.1.AND.NAME.EQ."ADJ") LAB(1)=NAME
                IF (JTIE.EQ.0.AND.NAME.EQ."BORDA") LAB(1)=NAME
                DO 215 J=1,NBR
                LIST(J)=JRANK(J)
            215 CONTINUE
                DO 216 J=2,NBR
            216 IF (SUMA(J).EQ.SUMA(J-1)) LIST(J)=-LIST(J)
                DO 217 I=1,NBR
                IF (JTIE.EQ.1.AND.NAME.EQ."BORDA") GO TO 217
                IF (JTIE.EQ.0.AND.NAME.EQ."BORDA") LISTC(I,1)=LIST(I)
                IF (JTIE.EQ.1.AND.NAME.EQ."ADJ") LISTC(I,1)=LIST(I)
                IF (JTIE.EQ.0.AND.NAME.EQ."ADJ") GO TO 217
                IF (NAME.EQ."PREF") LISTC(I,2)=LIST(I)
                IF (NAME.EQ."FUZZY")LISTC(I,3)=LIST(I)
            217 CONTINUE
                IF (NPRINT.EQ.5) RETURN
                DO 220 J=2,NBR
            220 IF( SUMA(J) .EQ. SUMA(J-1) )  JPREF(J) = " ="
                IF (NAME.EQ."ADJ") NAME="ADJ BORDA"
                WRITE(6,906)   NAME, ( JPREF(J), JRANK(J), J=1,NBR)
            906 FORMAT(//1XA10, (T12,24(A2,I3)//) )
                WRITE(6,908)
            908 FORMAT(1X)
                IF (NAME.EQ."ADJ BORDA") NAME="ADJ"
          C
                RETURN
                END
```

TABLE C-1. (CONTINUED)

```
SUBROUTINE PREF        74/74    OPT=1                                 FTN 4.6+439

          C                    COMPUTE PREFERENCE
          C
                SUBROUTINE  PREF
          C
                COMMON /CDATA/ NBR ,NJ,NWT, NAME(2,300),A(100,100),WI(300),WJ(101)
               *               , JNAME(101),NSIZE(101),JSUBL(100,101)
                COMMON/IDD/HEADER(8),NOTCOM,NPTYP1,NPTYP2,NFUZ,NPRINT,JTIE
                COMMON /WORK/X(100,100),SUMA(300),JRANK(100),JPREF(100)
                DIMENSION ZZETA(87),PP(87),NZETA(9)
          C
                DATA  NDASH/"------"/
                DATA PP/1.00,0.000,0.625,0.375,0.000,1.000,0.703,0.469,0.234,
               *0.117,0.000,1.000,0.773,0.509,0.398,0.208,0.120,0.051,0.022,0.000,
               *1.000,0.964,0.853,0.737,0.553,0.420,0.287,0.198,0.112,0.069,0.033,
               *0.017,0.006,0.002,0.000,1.000,0.949,0.859,0.768,0.679,0.520,0.399,
               *0.299,0.208,0.153,0.094,0.063,0.037,0.023,0.011,0.0064,0.0028,
               *0.0013,0.0004,0.0001,0.000,1.000,0.9976, 0.980,0.945,0.882,0.803,
               *0.702,0.611,0.498,0.408,0.320,0.248,0.183,0.138,0.095,0.057,0.045,
               *0.030,0.019,0.012,0.007,0.0042,0.0023,0.0013,0.0006,0.0003,
               *0.0001,0.0001,0.0001,0.0001,0.000/
                DATA ZZETA/0.000,1.000,0.000,0.500,1.000,0.000,0.200,0.400,0.600,
               *0.800,1.000,0.000,0.125,0.250,0.375,0.500,0.625,0.750,0.875,1.000,
               *0.000,0.072,0.143,0.214,0.286,0.357,0.429,0.500,0.572,0.643,0.715,
               *0.787,0.858,0.929,1.000,0.000,0.050,0.100,0.150,0.200,0.250,0.300,
               *0.350,0.400,0.450,0.500,0.550,0.600,0.650,0.700,0.750,0.800,0.850,
               *0.900,0.950,1.000,0.000,0.033,0.066,0.100,0.133,0.166,0.200,0.233,
               *0.266,0.300,0.333,0.366,0.400,0.433,0.466,0.500,0.533,0.566,0.600,
               *0.633,0.666,0.700,0.733,0.766,0.800,0.833,0.866,0.900,0.933,0.966,
               *1.000/
                DATA NZETA/0,0,1,3,6,12,21,36,57/
          C     SELECT TYPE OF CALCULATION
                EQVAL=0.5
                LTVAL=0.0
                ADVAL=1.0
                IF (NPTYP2.EQ.0) GO TO 90
                EQVAL=0.0
                LTVAL=-1.0
                ADVAL=0.0
             90 CONTINUE
          C
                DO 130 J=1,NBR
          C
                DO 120 I=J,NBR
                XIJ = A(I,J)
                XJI = A(J,I)
                X(I,J) = 0.0
                IF( I .EQ. J )  GO TO 120
                IF( XIJ .NE. XJI )  GO TO 110
                IF (XIJ.EQ.0.0 .AND.NPTYP1.EQ.0) GO TO 120
                X(I,J)=EQVAL
                X(J,I)=EQVAL
                GO TO 120
            110 V = 1.0
                IF (XIJ.LT.XJI) V=LTVAL
                X(I,J) = V
                X(J,I)=ADVAL-V
            120 CONTINUE
```

TABLE C-1. (CONTINUED)


```
      130 CONTINUE
    C              SUM A(I)
          DO 150 I=1,NBR
          SUMR = 0.0
          DO 140 J=1,NBR
      140 SUMR = SUMR + X(I,J)
      150 SUMA(I) = SUMR
    C
          IF (NPRINT.EQ.5) GO TO 170
          WRITE(6,902)  HEADER, ( J, J=1,NBR )
      902 FORMAT ("1COMPUTED PREFERENCE MATRIX",20X,8A10//"  SUM",(T13,19I6)
         *)
          WRITE(6,905) ( NDASH, J=1,NBR )
      905 FORMAT (1X,(T14,19A6))
          DO 160 I=1,NBR
      160 WRITE(6,904)  SUMA(I), I, ( X(I,J), J=1,NBR )
      904 FORMAT (1X,F6.1,I4,(T13,"I ",19F6.1))
      170 CONTINUE
    C
          CALL  ORDER( "PREF      " , NBR , SUMA )
          IF (NPRINT.EQ.5) RETURN
          IF (NPTYP2.EQ.0) GO TO 220
    C        TEMP          1 , 0 , -1
          XNM1 = NBR - 1
          DO 210 J=1,NBR
      210 SUMA(J) = 0.5 * ( SUMA(J) + XNM1 )
      220 CONTINUE
    C
    C        PROCEDURE FOR  ZETA
    C
    C     XX = NBR*(NBR-1)*(NBR+NBR-1)/12.0
          XX = NBR*(NBR-1)*(NBR-2)/6.0
          ZZ = 24.0 / ( NBR*NBR*NBR - NBR*( 4-3*MOD(NBR,2) ) )
    C        FIND NUMBER OF FRACTIONAL SUMS
          NF = 0
          DO 230 J = 1,NBR
          SA = JA = SUMA(J)
      230 IF( SA .NE. SUMA(J) )  NF = NF + 1
          IF( MOD( NF,2 ) .NE. 0 )  WRITE(6,910)  NF
      910 FORMAT (/"****ERROR**NUMBER OF FRACTIONAL SUMS =",I3/)
          WRITE (6,950) NF
      950 FORMAT (1X,"NUMBER OF FRACTIONAL SUMS=",I5)
          LE = 3
          IF( NF .EQ. 0 )  LE = 1
          DS = 0
          NR = 1
          NRC = 0
    C
          DO 310 L = 1,LE
          IF( L .EQ. 3 )  GO TO 305
          JSUM = 0
          I    = 0
          NH=2
          DO 300 J = 1,NBR
          SA = JA = SUMA(J)
          IF( SA .EQ. SUMA(J) )  GO TO 300
    C        FRACTION
```

```
            I = I + 1
            IF (I.EQ.NH) NR=NRC
            IF (I.EQ.NH) NH=NH+2
            JA = JA + NR
C     300   JSUM = JSUM + JA*JA
            NR=1
            IF (L.EQ.2) NR=0
            SA=JA
      300   JSUM = JSUM + JA*(JA-1)
C
            D     = XX - JSUM/2.0
            IF (D.LT.0.) WRITE (6,906)
      906   FORMAT (T130,"-")
            IF (D.LT.0.) D=0.
      305   ZETA = 1.0 - D * ZZ
C
C              TEST ZETA
C
            IF (NBR.GT.9) GO TO 25
            NSTART=NZETA(NBR)
       10   CONTINUE
            IF (ZETA.EQ.ZZETA(NSTART)) GO TO 30
            IF(ZETA.LT.ZZETA(NSTART+1).AND.ZETA.GT.ZZETA(NSTART)) GO TO 20
            NSTART=NSTART+1
            GO TO 10
C                 INTERPOLATION
       20   CONTINUE
            P=PP(NSTART+1)+(PP(NSTART)-PP(NSTART+1))*(ZETA-ZZETA(NSTART+1))/
           *(ZZETA(NSTART)-ZZETA(NSTART+1))
            GO TO 40
       30   P=PP(NSTART)
            GO TO 40
       25   CONTINUE
            GNU=(NBR*(NBR-1)*(NBR-2))/((NBR-4)**2)
            CS=(8./(NBR-4.))*(.25*(NBR*(NBR-1)*(NBR-2)/6.)-D+.5)+GNU
            PRINT 1, CS,GNU
        1   FORMAT (/1X,"CHI-SQUARE =",F10.3,5X,"DF ="F10.3)
            CALL MDCH (CS,GNU,P,IER)
            P=1.-P
       40   CONTINUE
            IF(  L .EQ. 1 )  LAB = "LOWER"
            IF(  L .EQ. 2 )  LAB = "UPPER"
            IF(  L .EQ. 3 )  LAB = "AVERAGE"
            IF( LE .EQ. 1 )  LAB = " "
            IF (D.LT.0.) WRITE(6,930) LAB   ,D,LAB
      930   FORMAT (/1X,"KENDALL D BRACKET",A10," D= ",F6.2, "THEREFORE ",A10,
           *"WILL BE ZERO")
            WRITE(6,908)  LAB, NBR, D, ZETA,  P
      908   FORMAT (/1X,A10," N= ",I3,5X,"KENDALL D = ",F10.2,4X,"ZETA = ",
           *F10.4,5X,"PROB THAT RANK ORDER NOT CONSISTANT = ",F10.8/)
            PTEST=.05
            DO 320 J=1,2
            KNOT="   "
            IF (P.GE.PTEST)KNOT="NOT"
            WRITE (6,920)KNOT,PTEST
            PTEST=.01
      320   CONTINUE
```

TABLE C-1. (CONTINUED)

```
SUBROUTINE PREF        /74    OPT=1                         FTN 4.6+439

          920 FORMAT (1X,"RANK ORDER ",A3," CONSISTANT AT",F4.2," LEVEL")
              NR=0
              NRC = 1
              DS = DS + D
              D = DS / 2.0
          310 CONTINUE
      C
      C
              RETURN
              END
```

TABLE C-1. (CONTINUED)

```
SUBROUTINE FUZZY      74/74    OPT=1                          FTN 4.6+439

      C          FUZZY RANK ORDER
      C
            SUBROUTINE FUZZY
      C
            COMMON /CDATA/ NBR ,NJ,NWT, NAME(2,300),A(100,100),WI(300),WJ(101)
           *               , JNAME(101),NSIZE(101),JSUBL(100,101)
            COMMON/IDD/HEADER(8),NOTCOM,NPTYP1,NPTYP2,NFUZ,NPRINT,JTIE
            COMMON /WORK/X(100,100),SUMA(300),JRANK(100),JPREF(100)
            DATA  NDASH/"------"/
      C
      C          DIVIDE EACH A(I,J) BY NUMBER OF JUDGES
            XNJ = NJ
            DO 110 J=1,NBR
            DO 110 I=1,NBR
        110 A(I,J) = A(I,J) / XNJ
      C
            WRITE(6,902) HEADER,  (J, J=1,NBR )
        902 FORMAT ("1NORMALIZED FREQUENCY MATRIX-FUZZY",14X,8A10//(T6,20I6))
            WRITE(6,904)  ( NDASH, J=1,NBR )
        904 FORMAT(T6,20A6)
            DO 120 I=1,NBR
        120 WRITE(6,906)  I, ( A(I,J), J=1,NBR )
        906 FORMAT (1X,I3,(T6,"I ",20F6.1))
      C
      C
            TRACE F = 0.0
            TRACE C = 0.0
            DO 280 I = 1,NBR
            DO 280 J = 1,NBR
            TRACE F = TRACE F  + A(I,J) * A(J,I)
            TRACE C = TRACE C  + A(I,J) * A(I,J)
        280 CONTINUE
      C          COMPUTE  F(R)
            FR = 2.0 * TRACE F / ( NBR*NBR - NBR )
      C          COMPUTE C(R)
            CR = 2.0 * TRACE C / ( NBR*NBR - NBR )
      C
            WRITE(6,912)  FR
            WRITE(6,914)  CR
        912 FORMAT(/" F(R) ="F7.3)
        914 FORMAT(/" C(R) =",F7.3)
      C
            DO 310 J = 1,NBR
            XX = 0.0
            DO 300 I = 1,NBR
            XX = AMAX1( XX , A(I,J)-A(J,I) )
        300 CONTINUE
            SUMA(J) = 1.0 - XX
        310 CONTINUE
      C
            WRITE(6,916)  (     J  , J=1,NBR )
            WRITE(6,918)  ( SUMA(J) , J=1,NBR )
        916 FORMAT(/// 1X,"PROJECT",(T12,20I5) )
        918 FORMAT(1X,"FUZZY RANK" ,(T14,20F5.2) )
      C
            CALL  ORDER( "FUZZY     " , NBR , SUMA )
      C
```

TABLE C-1. (CONTINUED)


```
      C
            RETURN
            END
```

TABLE C-1. (CONTINUED)

```
 SUBROUTINE CONCOR      74/74    OPT=1                              FTN 4.6+439

                  SUBROUTINE  CONCOR
            C
                  COMMON /CDATA/ NBR ,NJ,NWT, NAME(2,300),A(100,100),WI(300),WJ(101)
                 *              , JNAME(101),NSIZE(101),JSUBL(100,101)
            C
                  COMMON/IDD/HEADER(8),NOTCOM,NPTYP1,NPTYP2,NFU7,NPRINT,JTIE
                  REAL  XSUBL(100,101)
                  EQUIVALENCE  ( XSUBL,JSUBL )
            C
                  DIMENSION  R(101), JTEMP(100)
                  DIMENSION CCW(20,7,2)
                  DATA  NDASH/"------"/
                  DATA (CCW(J,3,1),J=1,20)/0.,7.8,18.,25.,31.,38.,43.,48.1,54.,60.,
                 *66.,71.9,77.8,83.8,89.8,95.8,102.,107.7,113.7,119.7/
                  DATA(CCW(J,4,1),J=1,20)/0.,18.5,35.,49.5,62.6,75.7,88.7,101.7,
                 *114.8
                 *,127.8,140.8,153.8,166.9,179.9,192.9,205.9,218.9,232.,245.,258./
                  DATA (CCW(J,5,1),J=1,20)/0.,37.,64.4,88.4,112.3,136.1,159.9,183.7,
                 *207.4,231.2,254.9,278.6,302.4,326.1,349.8,373.5,397.3,421.,444.8,
                 *468.5/
                  DATA (CCW(J,6,1),J=1,20)/0., 58.,103.9,143.3,182.4,221.4,260.2,
                 *299.,
                 *337.8,376.7,415.5,454.2,493.,531.7,570.5,609.3,648.1,686.8,725.6,
                 *764.4/
                  DATA (CCW(J,7,1),J=1,20)/0., 97.,157.3,217.,276.2,335.2,394.2,
                 *453.1,
                 *512.,571.,629.8,688.6,747.3,806.1,864.9,923.7,982.4,1041.2,1099.9,
                 *1158.7/
                  DATA (CCW(J,3,2),J=1,20)/0.,7.9,19.,31.,41.,52.,59.4,66.8,75.9,
                 *85.1,94.3,103.5,112.7,121.9,131.,140.2,149.4,158.6,167.8,177./
                  DATA (CCW(J,4,2),J=1,20)/0.,19.,42.,61.4,80.5,99.5,118.4,137.4,
                 *156.4,175.3,194.2,213.1,232.,250.9,269.8,288.7,307.6,326.4,345.3,
                 *364.2/
                  DATA (CCW(J,5,2),J=1,20 )/0.,39.,75.6,109.3,142.8,176.1,209.4,
                 *242.7,
                 *275.9,309.1,342.3,375.5,408.7,441.9,475.2,508.4,541.6,574.8,608.,
                 *641.2/
                  DATA (CCW(J,6,2),J=1,20)/0., 68.,122.8,176.2,229.4,282.4,335.4,
                 *388.3
                 *,441.2,494.,546.8,599.7,652.5,705.4,758.2,811.,863.8,916.6,969.4,
                 *1022.2/
                  DATA (CCW(J,7,2),J=1,20)/0.,104.,185.6,265.,343.8,422.6,501.2,
                 *579.9,
                 *658.4,737.,815.5,894.,972.5,1051.,1129.5,1208.,1286.5,1364.9,
                 *1443.4,1521.9/
            C
            C        NJ = NUMBER OF JUDGES
                  M = NJ
            C         NBR = NUMBER OF PROJECTS
                  N = NBR
                  NSUMT = 0
            C             GENERATE EVALUATION TABLE
                  DO 90 J = 1,M
            C        SIZE OF SUB-LIST
                  L = NSIZE(J)
            C                STORE SUB-LIST
                  DO 30 K = 1,L
```

TABLE C-1. (CONTINUED)

```
SUBROUTINE CONCOR      74/74    OPT=1                              FTN 4.6+439

         30 JTEMP(K) = JSUBL(K,J)
            XX = N + L + 1
            XX = XX / 2.0
      C                       COMPLETE SUB-LIST TABLE
            IF( L .EQ. N )  GO TO 45
            I = N - L
            NSUMT = NSUMT + I*I*I - I
            DO 40 K = 1,N
         40 XSUBL(K,J) = XX
      C
      C                        ASSUME NO MATCHES
         45 DO 50 K = 1,L
            JAV = IABS( JTEMP(K) )
            XX = K
         50 XSUBL(JAV,J) = XX
      C                        FIND MATCHES
            NEG = 0
            DO 80 K = 2,L
            JV = JTEMP(K)
            IF( JV .GT. 0 )  GO TO 60
            NEG = NEG + 1
            J2 = K
            IF( K .NE. L )  GO TO 80
         60 IF( NEG .LE. 0 )  GO TO 90
            J1 = J2 - NEG
            XX = J1 + J2
            XX = XX / 2.0
      C                      INSERT MATCHES
            DO 70 I = J1,J2
            JAV = IABS( JTEMP(I) )
         70 XSUBL(JAV,J) = XX
      C
            I = NEG + 1
            NSUMT = NSUMT + I*I*I - I
            NEG = 0
         80 CONTINUE
         90 CONTINUE
      C
            SUMT = NSUMT
            SUMT = SUMT / 12.0
      C
            WRITE(6,902)  HEADER, ( J, J=1,N )
        902 FORMAT (*1CONCORDANCE SUMMARY BY ELEMENT* 20X,8A10/* UNWEIGHTED SU
           *BLISTS*/(T12,20I6))
            WRITE(6,904)  ( NDASH, K=1,N )
        904 FORMAT(* JUDGE   *, (T13,20A6) )
      C
            DO 100 K=1,N
        100 R(K) = 0.0
            SUMR = 0.0
      C
            DO 110 J=1,M
            IF (JNAME(J).EQ.*ADJ*) JNAME(J)=*ADJ BORDA*
            WRITE(6,906)  JNAME(J), (XSUBL(K,J), K=1,N  )
        906 FORMAT (1XA10,(T13,*I *,20F6.1))
            DO 110 K=1,N
            R(K) = R(K) + XSUBL(K,J)
```

TABLE C-1. (CONTINUED)

```
SUBROUTINE CONCOR      74/74    OPT=1                          FTN 4.6+439

                SUMR = SUMR + XSURL(K,J)
        110 CONTINUE
      C
            RBAR = SUMR / N
            WRITE(6,908) NDASH, ( NDASH, K=1,N )
        908 FORMAT(T6,A5, (T13,20A6) )
            WRITE(6,910)   ( R(K), K=1,N )
        910 FORMAT(" R(J)",(T14,20 F6.1) )
      C
            S = 0.0
            DO 120 K=1,N
        120 S = S + ( R(K) - RBAR )**2
            WRITE(6,912)  RBAR, S
        912 FORMAT (/" MEAN =",F10.2,5X,"SUM OF DEVIATIONS SQUARED ="F10.2/)
            WRITE(6,914)  SUMT
        914 FORMAT(" SUM T =", F10.2)
      C
      C         KENDALS COEFFICIENT OF CONCORDANCE
            D=(M*M*(N*N*N-N)/12)-M*SUMT/12
            C=9999.
            IF (D.NE.0.) C=S/D
      C
            NDF = N - 1
      C
            CHISQ = M * NDF * C
            IF (C.EQ.9999.) WRITE (6,917)
        917 FORMAT (/1X,"KENDALLS COEFFICIENT OF CONCORDANCE IS INDETERMINATE"
           */)
      C
            WRITE(6,916)   C, M, N
        916 FORMAT(/ 1X,"KENDALLS COEFICIENT OF CONCORDANCE =",F10.3,5X,
           * "M ="I3,5X,"N ="I3/)
            IF (N.LE.7.AND.M.LE.20) GO TO 130
      C
            DF=NDF
            CALL MDCH (CHISQ,DF,P,IER)
            P=1.-P
            WRITE (6,918) CHISQ,NDF,P
        918 FORMAT(/" CHI-SQUARE =",F10.3,5X,"DF =",I4,5X,"P =",F6.4)
            PTEST=.05
            DO 320 J=1,2
            KNOT="   "
            IF (P.GE.PTEST)KNOT="NOT"
            WRITE (6,922)KNOT,PTEST
        922 FORMAT (1X,"RANK ORDER ",A3," CONSISTANT AT",F4.2," LEVEL.")
            PTEST=.01
        320 CONTINUE
            RETURN
      C
        130 CONTINUE
            TEST = .05
            PTEST=CCW(M,N,1)
            DO 325 J=1,2
            KNOT="   "
            IF (S.LE.PTEST) KNOT="NOT"
            WRITE (6,920) KNOT, TEST, PTEST
            TEST = .01
```

TABLE C-1. (CONTINUED)

```
SUBROUTINE CONCOR      74/74    OPT=1                         FTN 4.6+439

                  PTEST=CCW(M,N,2)
              325 CONTINUE
              920 FORMAT (1X,"RANK ORDER ",A3," CONSISTANT AT ",F4.2," LEVEL. CRITI
                 *CAL S = ",F7.2)
                  RETURN
                  END
```

TABLE C-1. (CONTINUED)

SUBROUTINE COMPARE   74/74   OPT=1   FTN 4.6+439

```
      SUBROUTINE COMPARE
      COMMON /CDATA/ NBR ,NJ,NWT, NAME(2,300),A(100,100),WI(300),WJ(101)
     *              , JNAME(101),NSIZE(101),JSUBL(100,101)
      COMMON/RANK/LISTC(100,3),LIST(100),LAB(3)
      DATA LAB/5H     ,4HPREF,5H     /
      NK=3
      NJ=2
      IF (LAB(3).NE."FUZZY") NK=2
      DO 10 K=1,NK
      DO 10 L=K,NK
      IF (K.EQ.L) GO TO 10
      JNAME(1)=LAB(K)
      JNAME(2)=LAB(L)
      NSIZE(1)=NBR
      NSIZE (2)=NBR
      DO 5 I=1,NBR
      JSUBL(I,1)=LISTC(I,K)
      JSUBL(I,2)=LISTC(I,L)
    5 CONTINUE
      CALL CONCOR
   10 CONTINUE
      LAB(1)="    "
      LAB(3)="     "
      RETURN
      END
```

TABLE C-1. (CONTINUED)


```
      SUBROUTINE REQUIRE
      COMMON/HELP/NV,IND(300),ICAT(300),NAM(2),JCHECK(300),ITT,IMAX,JMAX
      COMMON /CDATA/ NRR ,NJ,NWT, NAME(2,300),A(100,100),WI(300),WJ(101)
     *           , JNAME(101),NSIZE(101),JSUBL(100,101)
      DIMENSION IELEM(300,2),ISUM(300),JPROJ(300)
      DATA IR/0/
      IF (IR.NE.0) GO TO 55
      REWIND 9
      DO 50 I=1,300
      READ (9,100)(IELEM(I,J),J=1,2)
      IX=I
      IF (EOF(9).NE.0.) GO TO 40
  100 FORMAT(2A10)
      IR=1
   50 CONTINUE
      IX=301
   40 NELE=IX-1
      WRITE (6,950)
  950 FORMAT (////T10,*INDEX LIST OF REQUIREMENTS AND PROJECTS*)
      WRITE (6,101) (I,IELEM(I,1),IELEM(I,2),I=1,NELE)
  101 FORMAT (/4(I4,2X,2A10,*/*))
   55 IC=1
      II=0
      DO 65 I=1,IMAX
      ISUM(I)=0
   65 JPROJ(I)=0
      DO 11 I=1,NV
      K=IABS(JSUBL(I,NJ))+ITT
      IP=0
      N=1
      DO 10 J=1,NELE
      IF (NAME(1,K).EQ.IELEM(J,1)) GO TO 5
      GO TO 10
    5 II=II+1
      JPROJ(II)=IELEM(J,2)
      DO 15 M=1,ITT
   15 IF (JPROJ(II).EQ.NAME (1,M))ISUM(II)=IND(M)
      IF (JSUBL(I,NJ).LT.0.AND. N.EQ.1) ISUM(II)=-ISUM(II)
      IF (N.GT.1) ISUM(II)=-ISUM(II)
      N=N+1
      IP=IP+1
      IC=IC+1
   10 CONTINUE
      IF (IP.EQ.0)WRITE (6,900) NAME(1,K)
  900 FORMAT (/1X,A10,* IS NOT IN REQUIREMENT INDEX*/)
   11 CONTINUE
      DO 20 I=1,IC
      DO 20 J=I,IC
      IF (I.EQ.J) GO TO 20
      IF (IABS(ISUM(I)).NE.IABS(ISUM(J))) GO TO 20
      IF (ISUM(J).GT.0.AND.ISUM(J+1).LT.0) ISUM(J+1)=IABS(ISUM(J+1))
      ISUM(J)=0
   20 CONTINUE
      M=0
      DO 25 I=1,IC
   25 IF (ISUM(I).NE.0) M=M+1
   30 K=0
```

TABLE C-1. (CONCLUDED)

SUBROUTINE REQUIRE   74/74   OPT=1   FTN 4.6+439

```
          DO 35 I=1,IC
          IF (ISUM(I).EQ.0) GO TO 31
          GO TO 35
       31 IF (I.GT.M) GO TO 35
          DO 32 J=I,IC
          K=1
       32 ISUM(J)=ISUM(J+1)
       35 CONTINUE
          IF (K.EQ.1) GO TO 30
          NV=M
          NSIZE(NJ)=M
          DO 45 I=1,M
       45 JSUBL(I,NJ)=ISUM(I)
          RETURN
          END
```

REFERENCES

BOOKS

1. Arrow, K. J., Social Choice and Individual Values, New Haven, London: John Wiley, 1963.

2. Arrow, K. J., Selected Readings in Economic Theory From Econometrica, Cambridge, MA: MIT Press, pp. 302-313.

3. Aumann, Robert J., Human Judgments and Optimality, Chapter 12, New York: John Wiley, 1964.

4. Black, D., Committees and Elections, Cambridge, Great Britain: University Press, 1958.

5. Buchanan, J. and Tullock, G., The Calculus of Consent, Ann Arbor: University of Michigan, 1962.

6. Chamberlain, J. R., Essays on Public Goods, Externalities and Majority Rule, Stanford University Dissertation, 1974.

7. Cochrane, J. L. and Zeleny, M., Multiple Criteria Decision Making, University of South Carolina, 1973.

8. Cohon, J. L., Multiobjective Programming and Planning, New York: Academic Press, 1978.

9. Conover, W. J., Practical Nonparametric Statistics, New York: John Wiley, 1971.

10. Coombs, C. H., A Theory of Data, New York: John Wiley, 1964.

11. Crook, J. F., Separate Versus Concurrent Decision Procedures for Public Goods Provision Under Majority Rule, Virginia Polytech Institute Dissertation, 1973.

12. Delbeca, A. L., et al., Group Techniques for Program Planning, Glenview, Ill.: Scott, Foresman and Co., 1975.

13. Dobbins, Edward B., Research and Development Project Prioritization-An Annotated Bibliography, US Army Missile Command, Redstone Arsenal, Alabama 35809, Management Report RN-80-1, April 1980.

14. Dobbins, Edward B., Research and Development Prioritization - Computer Model, US Army Missile Command, Redstone Arsenal, Alabama 35809, Management Report RN-80-2, April 1980.

15. Edwards, A. L., Techniques of Attitude Scale Conservations, New York: Appleton Century Crafts, 1957.

16. Edwards, Allen L., Statistical Methods for the Behavioral Sciences, Chapter 19, New York: Holt, Rinehart, 1963.

17. Fishburn, P. C., The Theory of Social Choice, Princeton, New Jersey: Princeton University Press, 1973.

18. Haith, D. A., Studies in the Analysis of Metropolitan Water Resource Systems, Ithaca, New York: Cornell University, 1971.

19. Hollander, M. and Sethuraman, J., Testing for Agreement Between Two Groups of Judges, Florida State University, 1977.

20. Hanson, R. A., Majority Rule and Policy Outcomes: A Critique of the "Calculus of Consent," University of Minnesota Dissertation, 1973.

21. Hoyer, R. W., Some Multivariate Problems of a Spatial Model of Voting Under Majority Rule, a Virginia Polytechnic Institute Dissertation, 1976.

22. Kendall, M. G., Correlation Methods, New York: Hafner Publishing, 1962.

23. Kirkwood, C. W., Decision Analysis Incorporating Decisions of Groups, Massachusetts Institute of Technology, 1972.

24. Kyburg, H. E. and Smokler, H. E., Studies in Subjective Probability, New York: John Wiley, 1963.

25. Lazarsfeld, P. F. and Henry, N. W., Readings in Mathematical Social Science, Massachusetts Institute of Technology Press, 1966.

26. Lehmann, E. L. and D'Abera, H., Nonparametrics: Statistical Methods Based on Ranks, San Francisco, California: Holden Day, Inc., 1975.

27. Luce, D. R. and Raiffa, H., Games and Decisions, New York: John Wiley, 1957.

28. Pattanaik, P. K., Voting and Collective Choice-Some Aspects of the Theory of Group Decision-Making, Cambridge University, 1971.

29. Pattanaik, P. K., Strategy and Group Choice, New York: North Holland Publishing, 1978.

30. Raiffa, H., Decisions With Multiple Objectives: Preferences and Value Tradeoffs, New York: John Wiley, 1976.

31. Robbins, L., An Essay on the Nature Significance of Economic Science, London: MacMillan Press, 1935, 1972.

32. Rothenberg, J., The Measurement of Social Welfare, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1961.

33. Seaver, D. A., Assessment of Group Preferences and Group Uncertainty for Decision Making, Los Angeles: Social Science Research Institute, 1976.

34. Sen, A. K., Collective Choice and Social Welfare, San Francisco: Holden-Day, Inc., 1970.

35. Sheple, K. A., "Theories of Collective Choice", Political Science Annual, Vol. 5, New York: The Bobbs Merrill Company, 1974.

36. Shisko, R., A Survey of Solution Concepts for Majority Rule Games, Santa Monica: The Rand Corporation, 1974.

37. Siegel, S., Nonparametric Statistics for Behaviorial Sciences, New York: McGraw-Hill, 1956.

38. Singamsetti, N. R., Majority Rule, Specification and Policy Use of Social Preference Function, New York State University Dissertation, 1973.

39. Sticha, P. J., Tests of Single Coalition and Weighted Majority Theories of Group Decision Making, Michigan University Dissertation, 1977.

40. Svestka, J. A., The Development of a Rational Group Ordering Technique Without the Use of Cardinal Utilities, A Thesis, University of Alabama in Huntsville, Huntsville, Alabama, 1966.

41. Thrall, R. M. et al., Decision Processes, New York: John Wiley, 1954.

42. Thurstone, L. L., The Measurement of Values, Chicago: University of Chicago Press, 1959.

43. Tullock, C., Toward a Mathematic of Politics, Ann Arbor: University of Michigan Press, 1967.

44. Van den Bogaard, P. J. M. and Versluis, J., The Design of Socially Optimal Decision, New York: John Wiley, 1960.

45. Von Neumann, J. and Morgenstern, O., Theory of Games and Economic Behavior, Princeton, New Jersey: Princeton University Press, 1953.

46. Wendell, R. E. and Thorson, S. J., Some Generalizations of Social Decisions Under Majority Rule, Columbus: Ohio State University, 1973.

47. Wyatt, M. W., Amalgamating the Rankings of Judges With Various Degrees of Expertise, A Thesis, University of Alabama in Huntsville, Huntsville, Alabama, 1973.

PERIODICALS

48. Abrams, R., "The Voter's Paradox and the Homogeneity of Individual Preference Orders," Public Choice, Vol. 26, Summer 1976, pp. 19-27.

49. Allen, Glen O., "Beyond the Voter's Paradox," Ethics, Vol. 88, October 1977, pp. 50-61.

50. Armstrong, R. D., Cook, W. D., "Strategy Selection in Ordinal Ranking Problems," Research Report CCS 299, University of Texas, Austin: July 1977.

51. Arrow, K. J., "Rational Choice Functions and Orderings," Economica, Vol. 26, pp. 121-127, 1959.

52. Arrow, K. J., "Tullock and an Existence Theorem," Public Choice, Vol. 6, 1969, p. 105.

53. Bacharach, Michael, "Group Decisions in the Face of Differences of Opinion," Management Science, Vol. 22, October 1975, pp. 182-191.

54. Barbut, Marc, "Does the Majority Ever Rule?" Portfolio and Art New Annual, Vol. 4, 1961, pp. 16-83, 161-168.

55. Barton, David M., "Constitutional Choice and Simple Majority Rule: Comment," Journal of Political Economy, Vol. 81, 1973, pp. 471-480.

56. Bartoszynski, Robert, "Power Structure in Dichotomous Voting," Econometrica, Vol. 40, November 1972, pp. 1003-1019.

57. Bell, Colin E., "What Happens When Majority Rule Breaks Down?," Public Choice, Vol. 33, Issue 2, 1978, pp. 121-126.

58. Bergson, A., "A Reformulation of Certain Aspects of Welfare Economics," Quarterly Journal of Economics, Vol. 52, 1938, pp. 310-334.

59. Bezdek, James C. et al., "A Fuzzy Relations Space for Group Decision Theory," Fuzzy Sets and Systems 1, No. 4, October 1978, pp. 255-268.

60. Bezdek, James C. et al., "Fuzzy Relation Spaces for Group Decision Theory: An Application," Fuzzy Sets and Systems 2, No. 1, January 1979, pp. 5-14.

61. Birnbeig, J. G., Pondy, L. R., Davis, C. L., "Effect of Three Voting Rules on Resource Allocation Decisions," Management Science, Vol. 16, No. 6, February 1970, pp. 356-372.

62. Black, Duncan, "The Decisions of a Committee Using a Special Majority," Econometrica, Vol. 16, January 1948, pp. 245-261.

63. Black, Duncan, "On the Rationale of Group Decision Making," Jol of Political Economy, Vol. 36, February 1948, pp. 23-34.

64. Black D., "The Elasticity of Committee Decisions With An Altering Size of Majority," Econometrica, Vol. 16, 1948, pp. 262-270.

65. Black, D., "Partial Justification of the Borda Count," Public Choice, Vol. XXVIII, Winter 1976, pp. 1-15.

66. Blair, D. H., et al., "Impossibility Theorems Without Collective Rationality," Journal of Economic Theory, Vol. 13, December 1976, pp. 361-379.

67. Blau, J. H., "The Existence of Social Welfare Functions," Econometrica, Vol. 25, January 1957, pp. 302-313.

68. Blau, J. H. and Deb, Rajat, "Social Decision Functions and the Veto," Econometrica, Vol. 45, No. 4, May 1977, pp. 871-879.

69. Blin, J. M., et al., "Optimization Theory and Social Choice," Proc. of 6th Hawaiian Int. Conf. on Sys. Sci. Suppl. on Urban and Regional Systems: Modelling Analysis and Decision Making, pp. 158-161, 1973.

70. Blin, J. M., "Fuzzy Relations in Group Decision Theory," Journal of Cybernetics, Vol. 4, No. 2, April-June 1974, pp. 17-22.

71. Blin, J. M. and Whinston, A. B., "Fuzzy Sets and Social Choice," Journal of Cybernetics, Vol. 3, No. 4, 1974, pp. 28-36.

72. Blin, J. M. and Whinston, A. B., "A Note on Majority Rule Under Transitivity Constraints," Management Science, Vol. 20, No. 11, July 1974, pp. 1439-1440.

73. Blin, J. M. and Whinston, A. B., "Discriminant Functions and Majority Voting," Management Science, Vol. 21, January 1975, pp. 557-566.

74. Blin, J. M. and Satterthwaite, M. A., "Strategy-Proofness and Single-Peakedness," Public Choice, Vol. 26, Summer 1976, pp. 53-58.

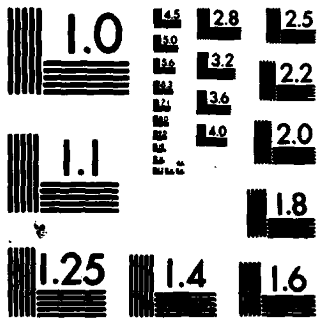
75. Blin, J. M. and Satterthwaite, M. A., "Individual Decisions and Group Decisions," Journal of Public Economics, Vol. 10, No. 2, 1978, pp. 247-267.

AD-A091 084 ARMY MISSILE COMMAND REDSTONE ARSENAL AL TECHNOLOGY--ETC F/G 5/1
A METHODOLOGY FOR AGGREGATION OF MULTIPLE CRITERIA RANK-ORDERED--ETC(U)
MAY 80 E B DOBBINS

UNCLASSIFIED DRSMI/RN-80-3-MR SBIE-AD-E950 034 NL

3

END
DATE
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

76. Bowman, V. J. and Colantoni, C. S., "The Extended Condorcet Condition: A Necessary and Sufficient Condition for the Transitivity of Majority Decision," Journal of Mathematical Sociology, Vol. 2, No. 2, July 1972, pp. 267-283.

77. Bowman, V. J. and Colantoni, C. S., "Transitive Majority Rule and the Theorem of the Alternative," Opns Res, Vol. 22, 1974, pp. 488-496.

78. Bowman, V. J. and Colantoni, C. S., "Majority Rule Under Transitivity Constraints," Management Science, Vol. 19, May 1973, pp. 1029-1041.

79. Bowman, V. J. and Colantoni, C. S., "Further Comments on Majority Rule Under Transitivity Constraints," Management Science, Vol. 20, 1974, p. 1441.

80. Brightwell, S. A., Cook, W. D., and Mehndiratta, S. L., "Assignment to Command Construction Projects," Info, Vol. 13(3), 1975, pp. 270-286.

81. Brown, D. J., "Aggregation of Preferences," Quarterly Journal of Econ. Theory, Vol. 89, No. 3, August 1975, pp. 456-469.

82. Buchanan, J. M., "Social Choice, Democracy, and Free Markets," Journal of Political Economy, Vol. 62, April 1954, pp. 114-123.

83. Buckles, B. P., "Two Applications of Fuzzy Set Theory to Decision Making," EG-799, Literature Survey, University of Alabama in Huntsville, Summer 1979.

84. Buckley, J. J. and Westen, T. E., "The Majority Rule Game," Int. J. Game Theory, Vol. 4, No. 1-2, 1975, pp. 105-112.

85. Buckley, J. J. and Westen, T. E., "Bargaining Set Theory and Majority Rule," Journal of Conflict Resolution, Vol. 20, No. 3, 1976, pp. 481-496.

86. Campbell, D. E., "Social Choice and Intensity of Preference," Journal of Political Economy, Vol. 81, 1973, pp. 211-218.

87. Campbell, Donald E., "Income Distribution Under Majority Rule and Alternative Taxation Criteria," Public Choice, Vol. 22, Summer 1975, pp. 23-35.

88. Campbell, Donald E., "Realization of Choice Functions," Econometrica, Vol. 46, 1978, p. 171.

89. Campbell, C. D. and Tullock, G., "A Measure of the Importance of Cyclical Majorities," Economic Journal, Vol. 75, December 1965, pp. 858-857.

90. Castore, Carl H., Peterson, Kevin, Goodrich, Thomas A., "Risky Shift: Social Value or Social Choice? An Alternative Model,"

Journal of Personality and Social Psychology, Vol. 20(3), December 1971, pp. 487-494.

91. Chartier, O. and Wertheimer, S., "Voting Simulation," 5th Annual Pittsburgh Conference on Modeling and Simulation, April 1974, pp. 24-26.

92. Chernous'ko, F. L., "Weight Factors in Expert Estimates," Cybernetics, No. 6, October 1972, pp. 1021-1024.

93. Cole, J. D. and Sage, A. P., "Multi-person Decision Analysis in Large Scale Systems - Group Decision Making," Franklin Inst. Jnl., Vol. 299, April 1975, pp. 245-268.

94. Colman, A. M. and Pountney, I., "Borda's Voting Paradox: Theoretical Likelihood and Electoral Occurrences," Behavioral Science, Vol. 23, 1978, pp. 15-20.

95. Cook, W. D. and Saipe, A. L., "Committee Approach to Priority Planning: The Median Ranking Method," Cahiers du Centre d'Etude de Rocherch Operationnelle, Vol. 18(3), 1973, pp. 337-350.

96. Cook, W. D. and Seiford, L. M., "Priority Ranking and Consensus Formation," Management Science, Vol. 24, No. 16, December 1978, p. 1732.

97. Dalkey, N., "An Impossibility Theorem for Group Probability Functions," Presented at NSF Conf. on Subjective Optimality, July 16-21, 1972, Rand P-4862.

98. Davis, J. M., "The Transitivity of Preferences," Behavioral Science, Vol. 3, 1958, pp. 26-33.

99. Davis, Otto, A., et al., "Social Preference Orderings and Majority Rule," Econometrica, Vol. 40, January 1972, pp. 147-157.

100. Davis, O. A., et al., "Social Preference Orderings and Majority Rule," Econometrica, Vol. 40, 1972, pp. 147-157.

101. De Grazia, Alfred, "Mathematical Derivation of an Election System," Isis, Vol. 44, June 1953, pp. 42-51.

102. DeMeyer, F. and Plott, C. R., "A Social Welfare Function Using 'Relative Intensity' of Preference," Quarterly Journal of Economics, Vol. 85, 1971, pp. 179-186.

103. Denzau, A. T., Parks, R. P., "Continuity of Majority-Rule Equilibrium," Econometrica, Vol. 43, 1975, pp. 858-866.

104. Dummett, Michael and Farquharson, Robin, "Stability in Voting," Econometrica, Vol. 29, January 1961, pp. 33-43.

105. Dutta, Bhaskar and Pattanaik, Prasanta, K., "On Nicely Consistent Voting Systems," Econometrica, Vol. 46, No. 1, January 1978, pp. 163-170.

106. Ehrenberg, A. S. C., "On Sampling From a Population of Rankers," Biometrica, Vol. 39, 1952, pp. 82-87.

107. Einhorn, H. and Hogarth, R., "Unit Weighting Schemes for Decision Making," Organizational Behavior and Human Performance, Vol. 13, 1975, pp. 171-192.

108. Farris, D. R. and Sage, A. P., "Introduction and Survey of Group Decision Making With Application to Worth Assessment," IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-5, No. 3, May 1975.

109. Feldman, A. M., "Very Unsubtle Version of Arrow's Impossibility Theorem," Econ. Inquiry, Vol. 12, December 1974, pp. 534-546.

110. Ferejohn, J. A. and Grether, D. M., "On a Class of Rational Social Decision Procedures," Journal of Econ. Theory, Vol. 8, 1974, pp. 471-482.

111. Ferejohn, J. A., "Decisive Coalitions in the Theory of Social Choice," Journal of Econ. Theory, Vol. 15, August 1977, pp. 301-306.

112. Ferejohn, J. and Page, T., "On the Foundations of Intertemporal Choice," Am. J. Agric. Econ., May 1978, pp. 269-275.

113. Fine, Kit, "Some Necessary and Sufficient Conditions for Representative Decision on Two Alternatives," Econometrica, Vol. 40, 1972, pp. 1083-1090.

114. Fishburn, P. C., "Preferences, Summation, and Social Welfare Functions," Management Science, Vol. 16, 1969, pp. 179-186.

115. Fishburn, P. C., "Arrow's Impossibility Theorem: Concise Proof and Infinite Voters," J. Econ. Theory, Vol. 2, 1970, pp. 103-106.

116. Fishburn, P. C., "Comments on Hansson's Group Preferences," Econometrica, Vol. 38, 1970, pp. 184-185.

117. Fishburn, P. C., "Intransitive Individual Indifference and Transitive Majorities," Econometrica, Vol. 38, No. 3, May 1970, pp. 482-489.

118. Fishburn, P. C., "The Irrationality of Transitivity in Social Choice," Behavioral Science, Vol. 15, 1970, pp. 119-123.

119. Fishburn, P. C., "A Comparative Analysis of Group Decision Methods," Behavioral Science, Vol. 16, 1971, pp. 538-544.

120. Fishburn, P. C., "Voter Concordance, Simple Majorities, and Group Decision Methods," Behavioral Science, Vol. 18, 1973, pp. 364-376.

121. Fishburn, P. C., "Aspects of One-Stage Voting Rules," Management Science, Vol. 21, 1974, pp. 422-427.

122. Fishburn, P. C., "On the Family of Linear Extensions of a Partial Order," J. Combin. Theory B (USA), Vol. 17, No. 3, 1974, pp. 240-3.

123. Fishburn, P. C., "Impossibility Theorems Without the Social Completeness Axiom," Econometrica, Vol. 42, No. 4, July 1974, pp. 695-704.

124. Fishburn, P. C., "Single-Peaked Preferences and Probabilities of Cyclical Majorities," Behavioral Sciences, Vol. 19, 1974, pp. 21-27.

125. Fishburn, P. C., "Simple Voting Systems and Majority Rule," Behavioral Sciences, Vol. 19, 1974, pp. 166-176.

126. Fishburn, P. C., "Three-Valued Representative Systems," Mathematical Systems Theory, Vol. 9, No. 3, 1975, pp. 265-280.

127. Fishburn, P. C., "Multicriteria Choice Functions Based on Binary Relations," Operations Research, Vol. 25, 1977, pp. 989-1012.

128. Fishburn, P. C., "A Strategic Analysis of Nonranked Voting Systems," SIAM J. On Appl. Math., Vol. 35, 1978, pp. 488-495.

129. Fishburn, P. C., and Gehrlin, W. V., "Borda's Rule Positionial Voting, and Condorcet's Simple Majority Principle," Public Choice, Vol. XXVIII, Winter 1976, pp. 79-88.

130. Fishburn, P. C. and Gehrlein, W. V., "Analysis of Simple Two Stage Voting Systems," Behavioral Science, Vol. 21, 1976, pp. 1-21.

131. Fishburn, P. C., Gehrlin, W. V., "Win Probabilities and Simple Majorities in Probabilistic Voting Situations," Mathematical Programming, Vol. 11, No. 1, 1976, pp. 28-41.

132. Friedman, M., "A Comparison of Alternative Tests of Significance for the Problem of M Rankings," Annals of Math. Statistics, Vol. II, 1940, p. 864.

133. Garman, Mark B., Kamien, Morton, I., "The Paradox of Voting: Probability Calculations," Behavioral Science, Vol. 13(4), 1968, pp. 306-316.

134. Gehrlein, William V., Fishburn, P. C., "Condorcet's Paradox and Anonymous Preference Profiles," Public Choice, Vol. 26, Summer 1976, pp. 1-18.

135. Gevers, Louis, "On Interpersonal Comparability and Social Welfare Orderings," Econometrica, Vol. 47, No. 1, January 1979, pp. 75-89.

136. Gibbard, Allan, "Manipulation of Schemes That Mix Voting With Chance," Econometrica, Vol. 45, No. 3, April 1977, pp. 665-681.

137. Gillett, Raphael, "Collective Indecision," Behavioral Science, Vol. 22, 1977, pp. 383-390.

138. Good, I. J., Tideman, T. N., "From Individual to Collective Ordering Through Multidimensional Attribute Space," Proc. R. Soc., Vol. 347, No. 1650, 1975, pp. 371-385.

139. Goodman, L. A. and Markowitz, H., "Social Welfare Functions Based On Individual Rankings," American Journal of Sociology, Vol. 58, 1952, pp. 257-262.

140. Grandmont, J. M., "Intermediate Preferences and the Majority Rule," Econometrica, Vol. 46, No. 2, 1978, pp. 317-330.

141. Greenberg, Joseph, "Consistent Majority Rules Over Compact Sets of Alternatives," Econometrica, Vol. 47, No. 3, May 1979, pp. 627-636.

142. Gustafson, D. H., Pai, G. K. and Cramer, G. C., "A Weighted Aggregate Approach to R & D Project Selection," AIIE Transactions, Vol. 3, March 1971, pp. 22-31.

143. Hamada, K., "Simple Majority Rule on Distribution of Income," Journal of Economic Theory, Vol. 6, No. 3, 1973, pp. 243-264.

144. Hansson, B., "Group Preferences," Econometrica, Vol. 37, 1969, pp. 50-54.

145. Hansson, B., "Voting and Group Decision Functions," Synthese, Vol. 20, 1969, pp. 526-537.

146. Harnett, D. L., "A Level of Aspiration Model for Group Decision Making," Journal of Personality and Social Psychology, Vol. 5, 1967, pp. 58-66.

147. Herzberger, H. G., "Ordinal Preference and Rational Choice," Econometrica, Vol. 41, 1973, pp. 187-237.

148. Hildreth, C., "Alternative Conditions for Social Orderings," Econometrica, Vol. 21, 1953, pp. 81-94.

149. Hinich, M. J., Ledyard, J. L. and Ordeshook, P. C., "Nonvoting and Existence of Equilibrium Under Majority Rule," Journal of Economic Theory, Vol. 4, No. 2, 1972, pp. 144-153.

150. Hoyer, R. W. and Mayer, L., "Social Preference Orderings Under Majority Rule," Econometrica, Vol. 43, July 1975, pp. 803-806.

151. Inada, Ken-Ichi, "On the Economic Welfare Function," Econometrica, Vol. 32, 1964, pp. 316-338.

152. Inada, K., "A Note on the Simple Majority Decision Rule," Econometrica, Vol. 32, October 1964, pp. 525-531.

153. Inada, K., "The Simple Majority Decision Rule," Econometrica, Vol. 37, 1969, pp. 490-506.

154. Jain, R., "A Procedure for Multiple-Aspect Decision Making Using Fuzzy Sets," Int. J. Systems Sci., Vol. 8, No. 1, 1977, pp. 1-7.

155. Jamison, D. and Luce, E., "Social Homogeneity and Probability of Intransitive Majority Rule," Journal of Economic Theory, Vol. 5, No. 1, 1972, pp. 79-87.

156. Jamison, D., "The Probability of Intransitive Majority Rule: An Empirical Study," Public Choice, Vol. 23, 1975, pp. 87-94.

157. Kaneko, M., "Necessary and Sufficient Conditions for Transitivity in Voting Theory," Journal of Economic Theory, Vol. 11, No. 3, 1975, pp. 385-393.

158. Keeney, R. L., "Group Preference Axiomatization With Cardinal Utility," Management Science, Vol. 23, October 1976, pp. 140-145.

159. Kelly, J. S., "Necessity Conditions in Voting Theory, Journal of Economic Theory, Vol. 8, No. 2, 1974, pp. 149-160.

160. Kemp, M. C. and Ng, Y. K., "On Existence of Social Welfare Functions, Social Ordering, and Social Decision Functions," Economica, Vol. 43, February 1976, pp. 59-66.

161. Kendall, M. G., "A New Measure of Rank Correlation," Biometrika, Vol. XXX, 1938, pp. 81-93.

162. Kendall, M. G. and Smith, B. B., "The Problem of M Rankings," Annals of Math Statistics, Vol. X, 1939, pp. 275-287.

163. Kendall, M. G. and Smith, B. B., "On the Method of Paired Comparisons, Biometrika, Vol. XXXI, 1939, pp. 324-345.

164. Klahr, D., "A Computer Simulation of the Paradox of Voting," American Political Science Review, Vol. 60, 1966, pp. 384-390.

165. Klee, A. J., "The Utilization of Expert Opinion," Amer. Inst. of Chemical Eng. Journal, Vol. 18, November 1972, pp. 1107-1115.

166. Koopmans, T. C., "On Flexibility of Future Preference," Human Judgments and Optimality, 1964, 243-254.

167. Kramer, G. H., "On A Class of Equilibrium Conditions for Majority Rule," Econometrica, Vol. 41, No. 2, 1973, pp. 285-297.

168. Kuga, K. and Nagatani, H., "Voter Antagonism and the Paradox of Voting," Econometrica, Vol. 42, No. 6, November 1974, pp. 1045-1067.

169. Little, I. M. D., "Social Choice and Individual Values," J. of Pol. Econ., Vol. 60, 1952, pp. 422-432.

170. Luce, R. D., "Semiorders and a Theory of Utility Discrimination," Econometrica, Vol. 24, No. 2, April 1956, pp. 272-285.

171. May, K. O., "A Set of Independent Necessary and Sufficient Conditions for Simple Majority Decision," Econometrica, Vol. 20, 1952, pp. 680-684.

172. May, K. O., "Intransitivity, Utility, and the Aggregation of Preference Patterns," Econometrica, Vol. 22, 1954, pp. 1-13.

173. McGuire, J. B. and Thompson, C. J., "On the Reconstruction of an Evolutionary Order," J. Theor. Bio., Vol. 75, 1978, pp. 141-147.

174. McKelvey, R. D., "Intransitivities in Multidimensional Voting Models and Some Implications for Agenda Control," Jour. of Economic Theory, Vol. 12, 1976, pp. 472-482.

175. McKelvey, R. D. and Ordeshook, P. C., "Symmetric Spatial Games Without Majority Rule Equilibria," Am. Pol. Sci. R., Vol. 70, December 1976, pp. 1172-1184.

176. McKelvey, R. D., Ordeshook, P. C., "An Undiscovered von Neumann-Morgenstern Solution for the (5, 3) Majority Rule Game," Int. Journal of Game Theory, Vol. 6, Nov. 1, 1977, pp. 33-34.

177. McKelvey, R. D., Wendell, R. E., "Voting Equilibria in Multi-dimensional Choice Spaces," Mathematics of Operations Research, Vol. 1, No. 2, May 1976, pp. 144-158.

178. Merchant, D. K. and Rao, M. R., "Majority Decisions and Transitivity: Some Special Cases," Management Science, Vol. 23, October 1976, pp. 125-130.

179. Monjardet, B., "Une Autre Preuve Du Theoreme D'Arrow," Recherche Operationnelle, Vol. 12, No. 3, 1978, pp. 291-296.

180. Moon, J. W., "A Problem on Rankings by Committees," Econometrica, Vol. 44, No. 2, March 1976, pp. 241-246.

181. Murakami, Y., "Some Logical Properties of Arrowian Social Welfare Function," The Journal of Economic Behavior, Vol. 1, No. 1 April 1961, pp. 77-84.

182. Murakami, Y., "Formal Structure of Majority Decision, Econometrica, Vol. 34, 1966, pp. 709-718.

183. Moran, P. A. P., "On the Method of Paired Comparisons," Biometrika, Vol. 34, 1947, pp. 363-365.

184. Nakamura, K., "The Core of a Simple Game With Ordinal Preferences," Int. Journal of Game Theory, Vol. 4, Issue 2, 1975, p. 95.

185. Navarrete, N. Jr., Fukushima, M., and Mine, H., "A New Ranking Method Based on Relative Position Estimate and Its Extensions," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-9, No. 11, November 1979, pp. 681-689.

186. Niemi, R. G. and Weisberg, H. F., "A Mathematical Solution for the Probability of the Paradox of Voting," Behaviorial Science, Vol. 13, 1968, p. 317.

187. Nitzan, Shmuel, "The Lexicographic Order, Majority Rule and Equilibrium," The Hebrew University of Jerusalem Department of Economics Research Report No. 56, May 1974, pp. 1-17.

188. Orlovsky, S. A., "Decision-Making With a Fuzzy Preference Relation," Fuzzy Sets and Systems 1, No. 3, 1978, pp. 155-167.

189. Ostojic, S., "Information Structure for Rational Decision Making in a Self-Management Economy," Third Int. Cong. of Cybernetics, Vol. 1, pp. 575-581.

190. Paris, D. C., "Plurality Distortion and Majority Rule," Behavioral Science, Vol. 20, 1975, pp. 125-135.

191. Parks, R. P., "Impossibility Theorem for Fixed Preferences," Rev. Economic Studies, Vol. 20, 1975, pp. 125-135.

192. Pattanaik, P. K., "Sufficient Conditions for the Existence of a Choice Set Under Majority Voting," Econometrica, Vol. 38, No. 1, January 1970, pp. 165-170.

193. Peleg, B., "Consistent Voting Systems," Econometrica, Vol. 46, No. 1, January 1978, pp. 153-161.

194. Pessemier, E. A. and Baker, N. R., "Project and Program Decisions in R & D," R & D Management, Vol. 2, October 1971, pp. 3-14.

195. Plott, C. R., "Axiomatic Social Choice Theory: An Overview and Interpretation," Am. J. Political Sci., Vol. 20, 1976.

196. Pomeranz, J. E., Weil, R. L., Jr., "Cylical Majority Problem," Commun. of ECM, Vol. 13, No. 4, April 1970, pp. 251-254.

197. Rader, Trout, "Induced Preferences on Trades When Preferences May Be Intransitive and Incomplete," Econometrica, Vol. 46, No. 1, January 1978, p. 137.

198. Rao, S. N., "On A Sufficient Condition for Transitivity of Majority Decision," American Economist, Fall 1972, pp. 90-92.

199. Riker, W., "Voting and the Summation of Preference: An Interpretative Bibliographical Review of Selected Developments During the Last Decade," Amer. Political Sci. Rev., Vol. 55, 1961, pp. 900-end.

200. Richelson, J. T., "A Comparative Analysis of Social Choice Functions," Behavioral Science, Vol. 20, 1975, pp. 331-342.

201. Richelson, J., "Conditions on Social Choice Functions, Public Choice, Vol. XXXI, Fall 1977, p. 79-110.

202. Richelson, J. T., "A Comparative Analysis of Social Choice Functions, II," Behavioral Science, Vol. 23, 1978, pp. 38-44.

203. Richelson, J. T., "A Comparative Analysis of Social Choice Functions, III," Behavioral Science, Vol. 23, 1978, pp. 169-176.

204. Rowse, G., Gustafson, D., Ludke, R., "Comparison of Rules For Aggregating Subjective Likelihood Ratios," Organizational Behavior and Human Perf., Vol. 12, 1974, pp. 274-285.

205. Rothenberg, Jerome, "Conditions For A Social Welfare Function," J. of Pol. Econ., Vol. 61, 1953, pp. 389-405.

206. Rubinstein, A., "A Note About The 'Nowhere Denseness' of Societies Having An Equilibrium Under Majority Rule," Econometrica, Vol. 47, No. 2, 1979, pp. 511-514.

207. Salles, M., Wendell, R. E., "A Further Result On The Core Of Voting Games," Int. J. Game Theory, Vol. 6, No. 1, 1977, pp. 35-40.

208. Samuelson, P., "Reaffirming The Existence of 'Reasonable' Bergson-Samuelson Social Welfare Functions," Economica, Vol. 44, February 1977, pp. 81-88.

209. Saposnik, R., "On Transitivity of Social Preference Relation Under Simple Majority Rule," Journal of Economic Theory, Vol. 10, No. 1, 1975, pp. 1-7.

210. Schuyler, G. L., "The Ordering of n Items Assigned to k Rank Categories by Votes of m Individuals," American Statistical Association Journal, Vol. 43, December 1948, pp. 559-563.

211. Schwartz, T., "Rationality And The Myth Of The Maximum," Nous, Vol. 6, 1972, pp. 97-117.

212. Sen, A. K., "Preferences, Votes And The Transitivity Of Majority Decisions," Economic Studies, Vol. 31, April 1964, pp. 163-165.

213. Sen, A. K., "A Possibility Theorem On Majority Decisions," Econometrica, Vol. 34, 1966, pp. 491-499.

214. Sen, A. K., "Interpersonal Aggregation and Partial Comparability," Econometrica, Vol. 38, No. 3, May 1970, pp. 393-409.

215. Sen, A. K., "Choice Functions and Revealed Preference," Review of Economic Studies, Vol. 38, 1971, pp. 307-317.

216. Sengupta, M., "On A Difficulty In The Analysis Of Strategic Voting," Econometrica, Vol. 46, No. 2, March 1978, pp. 331-343.

217. Shannon, R. E., "The Amalgamation Of Group Opinions Or Evaluations," Journal of Industrial Engineering, Vol. XIX, No. 6, June 1968, pp. xviii.

218. Shannon, R. E. and Biles, W. E., "The Utility of Certain Curriculum Topics To Operations Research Practitioners," Operations Research, Vol. 18, No. 4, July-August 1970, pp. 741-745.

219. Shepsle, K. A., "Institutional Arrangements and Equilibrium In Multidimensional Voting Models," American Journal Of Political Science, Vol. 23, No. 1, 1971, pp. 27-59.

220. Sheridan, T. B. and Sicherman, A., "Estimation Of A Group's Multiattribute Utility Function In Real Time By Anonymous Voting," IEEE Transactions On Systems, Man and Cybernetics, Vol. SMC-7, No. 5, May 1977, pp. 392-394.

221. Smith, J. H., "Aggregation of Preferences and Variable Electorate," Econometrica, Vol. 41, 1973, pp. 1027-1041.

222. Souder, W. E., "Achieving Organizational Consensus With Respect To R&D Project Selection Criteria," Management Science, Vol. 21, 1975, pp. 669-681.

223. Steinberg, Earle and Rinks, Dan, "An Application Of The Blin-Whinston Algorithm Resolving Fuzzy Group Preferences," Management Science, Vol. 25, No. 2, February 1979, pp. 159-170.

224. Taylor, M. J., "Graph-Theoretical Approaches To Theory Of Social Choice," Public Choice, Vol. 4, 1968, pp. 35-47.

225. Taylor, M., "Proof Of A Theorem On Majority Rule," Behavioral Science, Vol. 14(3), 1969, pp. 228-231.

226. Taylor, M., "Mathematical Political Theory," British Journal Of Political Science, Vol. 1, No. 3, July 1971, p. 340.

227. Theil, H., "On The Symmetry Approach To The Committee Decision Problem," Management Science, Vol. 9, 1963, pp. 380-393.

228. Tullock, G., "Constitutional Choice and Simple Majority Rule: Reply," Journal of Political Economy, Vol. 81, No. 2, 1973, pp. 480-484.

229. Ward, B., "Majority Voting and Alternative Forms of Public Enterprise," Second Conference In Urban Expenditure, 1963.

230. Watson, S. R., Weiss, J. J., and Donnell, M. L., "Fuzzy Decision Analysis," IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-9, No. 1, January 1979, pp. 1-9.

231. Weisberg, H. F. and Niemi, R. G., "A Pairwise Probability Approach To The Likelihood Of The Paradox Of Voting," Behavioral Science, Vol. 18, 1973, pp. 109-117.

232. Wendell, R. E., Thorson, S. J., "Some Generalizations of Social Decisions Under Majority Rule," Econometrica, Vol. 42, No. 5, 1974, pp. 893-912.

233. Whaley, C. P., "Fuzzy Decision-Making," Interface Age, November 1979, pp. 87-91.

234. Wilson, R. B., "Social Choice Theory Without Pareto Principle," Journal of Econ. Theory, Vol. 5, 1972, pp. 478-486.

235. Wilson, R. B., "On The Theory of Aggregation," Journal Of Econ. Theory, Vol. 10, 1975, pp. 89-99.

236. Winkler, R. L., "The Consensus of Subjective Probability Distributions," Management Science, Vol. 15, No. 2, October 1968, pp. B61-B75.

237. Wood, R., and Wilson, D. T., "Technique For Converting Ranks Into Measures," Brit. Journal of Psych., Vol. 68, August 1977, pp. 321-326.

238. Young, H. P., "An Axiomatization of Borda's Rule," Journal of Econ. Theory, Vol. 9, 1974, pp. 43-52.

239. Young, H. P., "A Note On Preference Aggregation," Econometrica, Vol. 42, 1974, pp. 1129-1131.

240. Young, H. P., "Social Choice Scoring Functions," SIAM J. Appl. Math., Vol. 28, June 1975, pp. 824-834.

241. Young, H. P. and Levenglick, A., "A Consistent Extension of Condorcet's Election Principle," SIAM J. of Appl. Math., Vol. 35, 1978, pp. 285-300.

DISTRIBUTION

	No. of Copies
Defense Technical Information Center Cameron Station Alexandria, VA 22314	12
IIT Research Institute ATTN: GACIAC 10 West 35th Street Chicago, IL 60616	1
Department of the Army ATTN: DAMA-AR, Dr. Lasser Washington, DC 20310	1
Commander US Army Materiel Development and Readiness Command ATTN: DRCDMD-ST, Dr. Wiseman	1
DRCLDC, Mr. J. Johnson 5001 Eisenhower Avenue Alexandria, VA 22333	1
Commander US Army Armament Research and Development Command ATTN: Technical Director Dover, NJ 07801	1
Commander US Army Aviation Research and Development Command ATTN: Technical Director St. Louis, MO 63166	1
Commander US Army Mobility Equipment Research and Development Command ATTN: Technical Director Fort Belvoir, VA 22060	1
Commander US Army Natick Research and Development Command ATTN: Technical Director Natick, MA 01760	1
Commander US Army Electronics Research and Development Command ATTN: Technical Director Adelphi, MD 20783	1

[redacted]

DISTRIBUTION (Continued)

Commander US Army Communications Research and Development Command ATTN: Technical Director Fort Monmouth, NJ 07703	1
Commander US Army Tank-Automotive Research and Development Command ATTN: Technical Director Warren, MI 48090	1
Commander US Army Materials and Mechanics Research Center ATTN: Technical Director Watertown, MA 02172	1
Commander Human Engineering Laboratory ATTN: Technical Director Aberdeen Proving Ground, MD 21005	1
Commander US Army Research Office P. O. Box 12211 Research Triangle Park, NC 27709	1
DRSMI-D	1
-DS	1
-R, Dr. Kobler	1
-RT	1
-RE	1
-RG	1
-RD	1
-RK	1
-RH	1
-RL	1
-RS	1
-RN, Dr. Dobbins	32
-RPR	3
-RPT, Record Copy	1

END

DATE
FILMED

8

DTIC